Parallel Evaluation of Algebraic Circuits

DISSERTATION
zur Erlangung des Grades eines Doktors
der Naturwissenschaften

vorgelegt von
M.Sc. Daniel König

Erstgutachter: Prof. Dr. Markus Lohrey
Zweitgutachter: Prof. Dr. Volker Diekert

Tag der Disputation: 25. Juli 2017

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen
Siegen 2017

Contents

Acknowledgment (in German)

Zuvorderst gilt mein großer Dank dem Betreuer dieser Arbeit, Prof. Dr. Markus Lohrey. Bereits bei unserer ersten Begegnung übergab er mir das Manuskript zu seinem Buch "The Compressed Word Problem for Groups", durch dessen Lektüre in mir eine große Neugierde für Algorithmen zur Auswertung komprimierter Strukturen geweckt wurde. Während meiner Zeit als Doktorand konnte ich mich jederzeit auf nötige Hilfe und Unterstützung verlassen. Sowohl menschlich als auch fachlich hätte ich mir keine bessere Betreuung vorstellen können. Ebenso möchte ich mich bei meinen Kollegen während dieser Zeit, Eric Nöth, Seungbum Jo, Philipp Reh und besonders bei meinen beiden Büromitbewohnern und Co-Autoren Danny Hucke und Moses Ganardi bedanken. Es war eine großartige Zeit mit vielen lustigen, unterhaltsamen aber auch stets lehrreichen Gesprächen. Ebenso danke ich Christoph Schlechtingen für Hilfe in allen technischen Bereichen. Mein Dank gilt ebenfalls den Mitgliedern der Promotionskommission Prof. Dr. Michael Möller, Prof. Dr. Volker Blanz und dem Zweitgutachter dieser Arbeit Prof. Dr. Volker Diekert.

Weiterhin gilt mein Dank den Menschen, ohne die ich erst gar nicht die Möglichkeit gehabt hätte, diese Dissertation zu schreiben: Zunächst meinen Eltern, die die Grundsteine für meinen Lebensweg gelegt haben, mir die nötige Freiheit gelassen haben, meinen Weg zu finden und mich dann doch hin und wieder durch leichte Korrekturen in die richtige Spur gebracht haben.

Anschließend meinen Lehrern, ohne die sich mein Interesse an der Mathematik nie so entwickelt hätte. Hier insbesondere Herrn Rausch für eine außergewöhliche und spannende Mathematik-AG, Herrn Scholl für einen hervorragenden Differenzierungskurs Mathe-Info, in dem viele mathematische Themen behandelt wurden, die im eigentlichen Mathematikunterricht zu kurz kommen. Durch diese zusätzlichen Angebote wurde mir der Übergang von der Schule zur Universität stark erleichtert und ich konnte mich von Anfang an von den Faszinationen der Mathematik fesseln lassen. Weiterhin danke ich Herrn Dartsch für einen stets unterhaltsamen und lehrreichen Mathe-Leistungskurs und Herrn Wolf (†), der den Beruf des Lehrers an der Grenze zur Perfektion auszuüben vermochte.

Mein Dank gilt ebenso allen Menschen, die mich durch mein Studium begleitet haben. Hier insbesondere meinen KomilitonInnen Julia Müller, Natalie Schmücker und Christian Jung. Ebenso danke ich Prof. Dr. Gregor Nickel und Dr. Theo Overhagen für hervorragende Einführungsvorlesungen in das Mathematikstudium. Weiterhin Prof. Dr. Bernd Dreseler für sehr spannende und Prof. Dr. Gerd Mockenhaupt für sehr besondere Vorlesungen kombiniert mit der außergewöhnlichen Fähigkeit, komplexeste mathematische Zusammenhänge verständlich zu erklären. Schließlich gilt mein Dank Prof. Dr. Dieter Spreen, der mein Interesse für die theoretische Informatik geweckt hat und mich hervorrangend während meiner beiden Abschlussarbeiten betreut hat.

Ein ganz großer Dank auch all meinen Freunden, vor allen denen, die mich schon seit der Kindergartenzeit begleiten und die ich nie missen möchte und an Lisa und Bert. Der letzte und größte Dank gilt meiner wunderbaren Frau Bianca, die mein Leben seit fast neun Jahren bereichert. Durch unser gemeinsames Leben, geprägt von starker gegenseitiger mentaler Unterstützung, blieben mir die sonst häufig mit dem Doktorandendasein einhergehenden Krisenzeiten vollständig erspart.

Abstract

The circuit evaluation problem for various finitely generated algebraic structures is studied. More precisely, for various rings, finite and infinite semirings, groups, and polynomial rings the complexity of circuit evaluation is investigated. The focus is on parallel complexity classes like NC or DET resp., the randomized parallel complexity class coRNC.

For the ring $(\mathbb{Z}, +, \cdot)$ it is known that circuit evaluation is in the randomized complexity class coRP. We show that under the assumption that the circuit has a constant bound for its multiplication depth circuit evaluation for $(\mathbb{Z}, +, \cdot)$ is complete for the class $\mathsf{C}_=\mathsf{L}$. If instead, we assume that the formal degree of the circuit is polynomially bounded, then circuit evaluation for $(\mathbb{Z}, +, \cdot)$ is complete for the class $\mathsf{C}_=\mathsf{LogCFL}$.

For circuits over the polynomial ring $(\mathbb{Z}[x_1, ..., x_k], +, \cdot)$, where k is part of the input, circuit evaluation is also known as polynomial identity testing and again the best known upper bound for this problem is coRP. Under the assumption that the circuit is skew, it is known that the problem is in coRNC. For skew circuits over $(\mathbb{Z}[x_1, ..., x_k], +, \cdot)$ with fixed k we show that circuit evaluation is in $\mathsf{C}_=\mathsf{L}$. The more general powerful skew circuits are introduced. These are skew circuits with variables where input gates can be labeled by powers x^n for binary encoded numbers n. It is shown that polynomial identity testing for powerful skew circuits belongs to coRNC^2 which generalizes the corresponding result for skew circuits. Two applications of this result are presented:
(i) Equivalence of higher-dimensional straight-line programs can be tested in coRNC^2; this result is even new in the one-dimensional case, where the straight-line programs produce strings.
(ii) The circuit evaluation problem for certain wreath products of finitely generated abelian groups belongs to coRNC^2.

For finitely generated linear groups, the best upper bound for circuit evaluation is again coRP, which was shown by a reduction to polynomial identity testing. Conversely, circuit evaluation for the linear group $\mathsf{SL}_3(\mathbb{Z})$ is equivalent to polynomial identity testing. In this work, it is shown that circuit evaluation for every finitely generated nilpotent group is in $\mathsf{DET} \subseteq \mathsf{NC}^2$. Within the larger class of polycyclic groups we find examples where circuit evaluation is at least as hard as polynomial identity testing for powerful skew circuits.

For finite semirings where semirings are not assumed to have an additive or multiplicative identity, the following dichotomy is shown: if a finite semiring is such that (i) the multiplicative semigroup is not solvable or (ii) it does contain a subsemiring with an additive identity 0 and a multiplicative identity $1 \neq 0$, then the circuit evaluation problem for the semiring is P-complete. In all other cases, the circuit evaluation problem is in DET.

An extension of the circuit evaluation problem to circuits over power sets is the circuit intersection problem where circuits with additional union gates over the power set of a structure are considered. We show that for a finite semiring S circuit intersection is in DET if S is a solvable local group. Otherwise it is P-complete. It is known that circuit intersection for the ring $(\mathbb{Z}, +, \cdot)$ is NEXPTIME-complete. We use this to show that also for the linear group $\mathsf{SL}_5(\mathbb{Z})$ circuit intersection is NEXPTIME-complete.

At the beginning of this thesis we show for the more classical word problem that this problem for an infinite finitely generated linear group G is DLOGTIME-uniform TC^0-complete if G is solvable and that it is in DLOGTIME-uniform NC^1 if G is virtually solvable.

Chapter 1

Introduction

Algebraic structures are the central objects of interest in the mathematical field of abstract algebra. But not only inside mathematics, also in other disciplines like quantum physics, chemistry and computer-science (especially in cryptography) these structures are of great interest. Given a formula over an algebraic structure, there are various ways to encode this formula into data. The most obvious way is to just "write down" the formula in the common way. Another idea is to encode formulas as tree graphs with labeled nodes. One can easily transform these encodings into each other. The encoding of our interest is a more succinct one: In this work so-called circuits over algebraic structures are investigated. This means the formula is encoded by a directed acyclic graph. This encoding can be seen as a kind of compression of the formula with a possibly exponential large compression ratio. In this thesis, the task to evaluate these circuits (without decompressing them) is investigated. The study of circuit evaluation problems has a long tradition in theoretical computer science and is tightly connected to many aspects in computational complexity theory. In its most general formulation, one has an algebraic structure $\mathcal{A} = (A, f_1, \ldots, f_k)$ where the f_i are mappings $f_i : A^{n_i} \to A$. A circuit over the structure $\mathcal{A}$ is a directed acyclic graph (dag) where every inner node is labeled with one of the operations f_i and has exactly n_i incoming edges that are linearly ordered. For the structures considered here, the leaf nodes of the dag are labeled with elements from a finite generating set of $\mathcal{A}$, and there is a distinguished output node. The task is for two given circuits to test, whether the output nodes evaluate to the same element. Ladner [52] proved that the circuit evaluation problem for the Boolean semiring $\mathbb{B}_2 = (\{0, 1\}, \vee, \wedge)$ is P-complete. This result marks a cornerstone in the theory of P-completeness [40], and motivated the investigation of circuit evaluation problems for other algebraic structures.

Another of the most important circuit evaluation problems is *polynomial identity testing*. Here, for some unitary ring R the input is a circuit over the structure $(R[x_1, \ldots, x_k], +, \cdot)$ whose internal gates are labeled with either addition or multiplication, its input gates are labeled with variables $(x_1, x_2, \ldots, x_k)$ or constants $(-1, 0, 1)$, and it is asked whether the output gate evaluates to the zero polynomial. Based on the Schwartz-Zippel-DeMillo-Lipton Lemma ([33], [78], [93]), Ibarra and Moran [45] proved that polynomial identity testing for $R = \mathbb{Z}$ belongs to the class coRP (the complements of problems that can be solved in randomized polynomial time). Whether there is a deterministic polynomial time algorithm for polynomial identity testing is an important problem that has implication for some major open problems in complexity theory. Although it is quite plausible that polynomial identity testing belongs to P (by [46]), it will be probably very hard to prove (by [48]).

It is known that for algebraic formulas (where the circuit is a tree) and for skew algebraic circuits (where for every multiplication gate, one of its two input gates is a constant or a variable), polynomial identity testing for $R = \mathbb{Z}$ belongs to coRNC (the complements of problems that can be solved randomized in polylogarithmic time by polynomially many processors in parallel), but it is still not known to be in P, see [48, Corollary 2.1]. This holds, since algebraic formulas and skew algebraic circuits can be evaluated in NC if the variables are substituted by concrete (binary coded) numbers. Then, as for general polynomial identity testing, the Schwartz-Zippel-DeMillo-Lipton

Lemma yields a coRNC-algorithm.

In Chapter 6 we identify a larger class of algebraic circuits, for which polynomial identity testing for $R = \mathbb{Z}$ or $R = \mathbb{Z}_p$ for a prime p is in coRNC; we call these circuits *powerful skew circuits*. In such a circuit, we require that for every multiplication gate, one of its two input gates is either a constant or a power x^N of a variable x where the exponent N is given in binary notation. One can replace this power x^N by a subcircuit of size $\lceil \log N \rceil$ using iterated squaring, but the resulting circuit is no longer skew. As mentioned above, the main result of Chapter 6 states that polynomial identity testing for powerful skew circuits over the rings $\mathbb{Z}$ and $\mathbb{Z}_p$ for p prime is still in coRNC. To show this, we use an identity testing algorithm of Agrawal and Biswas [4], which computes the output polynomial of the circuit modulo a polynomial $p(x)$ of polynomially bounded degree, which is randomly chosen from a certain sample space. Moreover, in our application, all computations can be done in the ring $\mathbb{F}_p[x]$ for a prime number p of polynomial size. This allows us to compute the big powers x^N modulo $p(x)$ in NC^2 using an algorithm of Fich and Tompa [38]. It should be noted that the application of the Agrawal-Biswas algorithm is crucial in our situation. If, instead we would use the Schwartz-Zippel-DeMillo-Lipton Lemma, then we would be forced to compute $a^N \mod m$ for randomly chosen numbers a and m with polynomially many bits. Whether this problem (modular powering) belongs to NC is a famous open problem [40, Problem B.5.6].

In Section 6.3 we present an application of our coRNC identity testing algorithm. It concerns the equivalence problem for straight-line programs. Here, a straight-line program (SLP) is a context-free grammar G that computes a single word $\mathsf{val}(G)$. In this context, SLPs are extensively used in data compression and algorithmics on compressed data, see [55] for an overview. It is known that equivalence for SLPs, i.e., the question whether $\mathsf{val}(G) = \mathsf{val}(H)$ for two given SLPs, can be decided in polynomial time. This result was independently discovered by Hirshfeld, Jerrum, and Moller [44], Mehlhorn, Sundar, and Uhrig [63], and Plandowski [72]. All known algorithms for the equivalence test are sequential and it is not clear how to parallelize them. Here, we exhibit an NC^2-reduction from the equivalence problem for SLPs to identity testing for powerful skew circuits. Hence, equivalence for SLPs belongs to coRNC. Moreover, our reduction immediately generalizes to higher dimensional pictures for which SLPs can be defined in a fashion similar to the one-dimensional (string) case, using one concatenation operation in each dimension. For two-dimensional SLPs, Berman et al. [21] proved that equivalence belongs to coRP using a reduction to polynomial identity testing over $\mathbb{Z}_2$. We can improve this result to coRNC. Whether equivalence of two-dimensional (resp., one-dimensional) SLPs belongs to P (resp., NC) is open.

Starting with [54] the circuit evaluation problem has been also studied for infinite finitely generated (f.g) monoids, in particular infinite f.g. groups. In this context, the input gates of the circuit are labeled with generators of the monoid and the internal gates compute the product of the two input gates. There and in subsequent work, the circuit evaluation problem is also called *compressed word problem*.

The classical *word problem* for a f.g. monoid M asks whether two given words over the alphabet of monoid generators evaluate to the same element of M. In case M is a group, this problem is equivalent to the question whether a given word over the generators evaluates to the identity element of the group. In Chapter 4 we show that the classical word problem for finitely generated linear solvable groups is in DLOGTIME-uniform TC^0.

If we consider a multiplicative circuit over a f.g. monoid M, then one can evaluate the circuit also in the free monoid Γ^* where Γ is the set of monoid generators that appear at the input gates of the circuit. The result will be a word over Γ, whose length can be exponential in the number of circuit gates. Hence, the circuit can be seen as a compressed representation of the word it produces. Formally, the compressed word problem for the monoid M asks for two given circuits whether the words they produce evaluate to the same element (or, in case of a group, whether a single circuit produces a word that evaluates to the group identity).

To avoid confusion, we will maintain the terminology circuit evaluation problem even in the case that the circuit is evaluated over a monoid resp. a group. One of the main motivations for the circuit evaluation problem for groups is the fact that the classical word problem for certain groups (automorphism groups, group extensions) can be reduced to the circuit evaluation problem

for simpler groups [57, Section 4.2]. For infinite groups the circuit evaluation problem was studied for the first time in [54]. Subsequently, several important classes of f.g. groups with polynomial time circuit evaluation problems were found: f.g. nilpotent groups, f.g. free groups, graph groups (also known as right-angled Artin groups or partially commutative groups), and virtually special groups. The latter contain all Coxeter groups, one-relator groups with torsion, fully residually free groups, and fundamental groups of hyperbolic 3-manifolds; see [57] for details. For the important class of f.g. linear groups, i.e., f.g. groups of matrices over a field, one can show that the circuit evaluation problem reduces to polynomial identity testing over $\mathbb{Z}$ or $\mathbb{Z}_p$ (depending on the characteristic of the field), and hence belongs to coRP [57, Theorem 4.15]. Vice versa, it was shown that polynomial identity testing over $\mathbb{Z}$ can be reduced to circuit evaluation for the linear group $\mathsf{SL}_3(\mathbb{Z})$ [57, Theorem 4.16]. The proof is based on a construction of Ben-Or and Cleve [20]. This result indicates that derandomizing the circuit evaluation problem for a f.g. linear group will be in general very difficult.

As a second application of the result that polynomial identity testing over $\mathbb{Z}$ for powerful skew circuits is in coRNC we consider in Section 7.2 the circuit evaluation problem for wreath products. If G is a f.g. non-abelian group, then circuit evaluation for the wreath product $G \wr \mathbb{Z}$ is coNP-hard [57, Theorem 4.21]. On the other hand, we prove that $\mathsf{CEP}(\mathbb{Z} \wr \mathbb{Z})$ is equivalent w.r.t. NC^2-reductions to polynomial identity testing for powerful skew circuits over $\mathbb{Z}$. In particular, $\mathsf{CEP}(\mathbb{Z} \wr \mathbb{Z})$ belongs to coRNC. The latter result generalizes to any wreath product $G \wr H$ where $H = \mathbb{Z}^n$ for some n and G is a finite direct product of copies of $\mathbb{Z}$ and $\mathbb{Z}_p$ for primes p.

Subsequently we further investigate the tight correspondence between circuits over commutative rings and circuits over non-commutative linear groups. In Section 7.3 we study the complexity of the circuit evaluation problem for f.g. nilpotent groups. For these groups, circuit evaluation can be solved in polynomial time [57]. Here, we show that for every f.g. nilpotent group the circuit evaluation problem belongs to the parallel complexity class $\mathsf{DET} \subseteq \mathsf{NC}^2$, which is the class of all problems that are AC^0-reducible to the computation of the determinant of an integer matrix, see [28, 30]. To the knowledge of the author, f.g. nilpotent groups are the only examples of infinite groups for which the circuit evaluation problem belongs to NC. Even for free groups, circuit evaluation is P-complete [54]. The main step of our proof is to show that for a torsion-free f.g. nilpotent group G the circuit evaluation problem belongs to the logspace counting class $\mathsf{C}_=\mathsf{L}$ (and is in fact $\mathsf{C}_=\mathsf{L}$-complete if G is nontrivial). To show this, we use the well-known fact that a f.g. torsion-free nilpotent group can be embedded into the group $\mathsf{UT}_d(\mathbb{Z})$ of d-dimensional unitriangular matrices over $\mathbb{Z}$ for some fixed d. Then, circuit evaluation for $\mathsf{UT}_d(\mathbb{Z})$ is reduced to the question whether two additive circuits over the natural numbers evaluate to the same number, which is $\mathsf{C}_=\mathsf{L}$-complete. There are several $\mathsf{C}_=\mathsf{L}$-complete problems related to linear algebra [5].

We also study the circuit evaluation problem for the matrix group $\mathsf{UT}_d(\mathbb{Z})$ for the case that the dimension d is not fixed, i.e., part of the input (Section 7.4). In this case, circuit evaluation turns out to be complete for the counting class $\mathsf{C}_=\mathsf{LogCFL}$, which is the LogCFL-analogue of $\mathsf{C}_=\mathsf{L}$.

In Section 7.6 we move from nilpotent groups to polycyclic groups. These are solvable groups where every subgroup is finitely generated. By results of Auslander and Swan [16, 81] these are exactly the solvable subgroups of $\mathsf{GL}_d(\mathbb{Z})$ for some d. We prove that polynomial identity testing over $\mathbb{Z}$ for powerful skew circuits reduces to the circuit evaluation problem for a specific 2-generator polycyclic group of Hirsch length three. As mentioned above even for skew circuits, no polynomial time algorithm is currently known (although the problem belongs to coRNC^2), see for instance [13, p. 6].

A large part of the literature about circuit evaluation is focused on arithmetic (semi)rings like $(\mathbb{Z}, +, \cdot)$, $(\mathbb{N}, +, \cdot)$ or the max-plus semiring $(\mathbb{Z} \cup \{-\infty\}, \max, +)$ [7, 51, 68, 69, 85]. These papers mainly consider semirings of polynomial formal degree. For commutative semirings, circuits of polynomial formal degree can be restructured into an equivalent (unbounded fan-in) circuit of polynomial size and logarithmic depth [85]. This result leads to NC-algorithms for evaluating polynomial degree circuits over commutative semirings [68, 69]. Over non-commutative semirings, circuits of polynomial formal degree do in general not allow a restructuring into circuits of logarithmic depth [51].

In [69] it was shown that also for finite non-commutative semirings circuit evaluation is in NC

for circuits of polynomial formal degree. On the other hand, the author is not aware of any NC-algorithms for evaluating general (exponential degree) circuits over semirings. The lack of such algorithms is probably due to Ladner's P-completeness result, which seems to exclude any efficient parallel algorithm (unless P = NC). On the other hand, in the context of semigroups, there exist NC-algorithms for circuit evaluation. In [19], the following dichotomy result was shown for finite semigroups: If the finite semigroup is solvable (meaning that every subgroup is a solvable group), then circuit evaluation is in NC (in fact, in DET), otherwise circuit evaluation is P-complete.

In Chapter 8, we extend the work of [19] from finite semigroups to finite semirings. On first sight, it seems that again Ladner's result excludes efficient parallel algorithms: It is not hard to show that if the finite semiring has an additive identity 0 and a multiplicative identity $1 \neq 0$ (where 0 is not necessarily absorbing with respect to multiplication), then circuit evaluation is P-complete. Therefore, we take the most general reasonable definition of semirings: A semiring is a structure $(R, +, \cdot)$ where $(R, +)$ is a commutative semigroup, $(R, \cdot)$ is a semigroup, and $\cdot$ distributes (on the left and right) over $+$. In particular, we neither require the existence of a 0 nor a 1. Our main result states that in this general setting there are only two obstacles to efficient parallel circuit evaluation: Non-solvability of the multiplicative structure and the existence of a zero and a one (different from the zero) in a subsemiring. More precisely, we show the following two results, where a semiring is called $\{0, 1\}$-free if there exists no subsemiring in which an additive identity 0 and a multiplicative identity $1 \neq 0$ exist:

(1) If a finite semiring is not $\{0, 1\}$-free, then the circuit evaluation problem is P-complete.

(2) If a finite semiring $(R, +, \cdot)$ is $\{0, 1\}$-free, then the circuit evaluation problem for $(R, +, \cdot)$ can be solved with AC^0-circuits that are equipped with oracle gates for (a) graph reachability, (b) the circuit evaluation problem for the commutative semigroup $(R, +)$ and (c) the circuit evaluation problem for the semigroup $(R, \cdot)$.

Together with the dichotomy result from [19] (and the fact that commutative semigroups are solvable) we get the following result: For every finite semiring $(R, +, \cdot)$, the circuit evaluation problem is in NC (in fact, in DET) if $(R, \cdot)$ is solvable and R is $\{0, 1\}$-free. Moreover, if one of these conditions fails, then circuit evaluation is P-complete.

The hard part of the proof is to show the above statement (2). To show this statement, we proceed in two steps. First we show that circuit evaluation for $\{0, 1\}$-free semirings can be reduced to so-called type-admitting circuits defined in Section 8.2 . We then come up with a parallel algorithm that evaluates such type-admitting circuits. The algorithm uses a rank-function on the semiring (the definition of a rank-function is given in Section 8.2.2). This algorithm reduces the type-admitting circuit by iteratively evaluating purely additive subcircuits and purely multiplicative subcircuits. The existence of the rank-function ensures that after a constant number of iterations, the whole circuit is evaluated. We then construct a rank-function with the required conditions for every finite $\{0, 1\}$-free semiring.

The above observation leads also to results about circuits over power structures of a finite semigroup $(S, \cdot)$, i.e., circuits with union gates and semigroup operation gates where the operation is evaluated on sets via $A \cdot B = \{a \cdot b \mid a \in A, b \in B\}$. We see in Chapter 9 that in the case S is a finite local group, the question whether the intersection of the represented sets of two circuits over the power semiring $\mathcal{P}(S) := (2^S \setminus \emptyset, \cup, \cdot)$ is non-empty (we call this the circuit intersection problem for S) can be reduced to circuit evaluation for S. In the case S is not a local group it is P-complete. In contrast to that fact, in [62] was shown that the circuit intersection problem for $(\mathbb{Z}, +)$ is NP-complete, while we know that circuit evaluation for $(\mathbb{Z}, +)$ is $\mathsf{C}_=\mathsf{L}$-complete. Finally in Section 9.4 a related result is shown. By [57] we know that circuit evaluation for $\mathsf{SL}_5(\mathbb{Z})$ is equivalent to polynomial identity testing over $\mathbb{Z}$ and so in coRP. We show that the circuit intersection problem for $\mathsf{SL}_5(\mathbb{Z})$ is NEXPTIME-complete. So here again, in contrast to the finite case, we get a massive raise of complexity.

In the following section the required complexity classes are defined and useful properties are shown. After that we present some facts about algebraic structures. We define the structures that will be investigated (semigroups, monoids, (finitely generated) groups and semirings) and

show some important properties of these structures. As mentioned above we then show a new result about the word problem for finitely generated linear solvable groups. After that the circuit evaluation problem is defined and we start to show the new results for this problem.

Chapter 2

Computational complexity

For a deeper background in complexity theory the reader might consult [11]. We assume that the reader is familiar with the standard complexity classes L (deterministic logarithmic space), NL (nondeterministic logarithmic space), P (deterministic polynomial time), NP (nondeterministic polynomial time), PSPACE (polynomial space) and $\mathsf{NEXPTIME}$ (nondeterministic exponential time). P-hardness will refer to logspace-reductions.

2.1 Circuit complexity classes

We use standard definitions concerning circuit complexity, see e.g. [87]. We only consider polynomially bounded families $(C_n)_{n\geq 0}$ of Boolean circuits where the number of gates of C_n is bounded by a polynomial $p(n)$. For such a family, gates of C_n can be encoded with bit strings of length $O(\log n)$. We will consider the class TC^0 of all problems that can be recognized by a polynomial size circuit family of constant depth built up from NOT-gates (which have fan-in one) and AND-gates, OR-gates and MAJORITY-gates (i.e., gates that return 1 if and only if more than the half of its inputs are 1) of unbounded fan-in. If MAJORITY-gates are not allowed, we obtain the class AC^0. The class NC^k $(k \geq 1)$ is defined by polynomial size circuit families of depth $O(\log^k n)$ that use NOT-gates, and AND- and OR-gates of fan-in two. One defines $\mathsf{NC} = \bigcup_{k\geq 1} \mathsf{NC}^k$. A family of AC^0- resp. TC^0-circuits $(C_n)_{n\geq 0}$ is $\mathsf{DLOGTIME}$-uniform, if for given binary coded gates u, v of C_n, one can (i) compute the type of gate u in time $O(\log n)$ and (ii) check in time $O(\log n)$ whether u is an input gate for v. Note that the time bound $O(\log n)$ is linear in the input length $|u| + |v|$. To define $\mathsf{DLOGTIME}$-uniformity for NC^1-circuits one needs the so-called extended connection language. We will not define this in detail, since we will not work with uniformity explicitly (for details consider e.g. [29]). For $k \geq 2$ $\mathsf{DLOGTIME}$-uniformity for NC^k-circuits is equivalent to logspace-uniformity, which means that the n-th circuit in the family can be computed in logarithmic space from the unary encoding of n. All circuit families in this paper are implicitly assumed to be $\mathsf{DLOGTIME}$-uniform. The above language classes can be easily generalized to classes of functions by allowing circuits with several output gates. Of course, this only allows to compute functions $f : \{0,1\}^* \to \{0,1\}^*$ such that $|f(x)| = |f(y)|$ whenever $|x| = |y|$. If this condition is not satisfied, one has to consider a suitably padded version of f.

The following result was shown in [35, 43]:[1]

Theorem 2.1. *The following problems belong to* $\mathsf{DLOGTIME}$*-uniform* TC^0 *for* $R = \mathbb{Z}$ *with binary encoded coefficients and* $R = \mathbb{F}_p$*:*

Input: *Polynomials* $p_1(x), \ldots, p_n(x) \in R[x]$.
Task: *Compute* $\prod_{i=1}^{n} p_i(x)$.

[1] Explicitly, the result is stated in [43, Corollary 6.5], where the authors note that Eberly's reduction [35] from iterated polynomial multiplication to iterated integer multiplication is actually an AC^0-reduction, which yields a $\mathsf{DLOGTIME}$-uniform TC^0 bound with the main result from [43].

Input: *Two polynomials* $p(x), q(x) \in R[x]$.
Task: *Compute* $r(x)$ *and* $s(x)$ *with* $\deg(r) < \deg(q)$ *such that* $p(x) = q(x) \cdot s(x) + r(x)$.

We use the standard notion of constant depth Turing-reducibility: for functions $f_1, \ldots, f_k$ let $\mathsf{AC}^0(f_1, \ldots, f_k)$ be the class of all functions that can be computed with a polynomial size circuit family of constant depth that uses NOT-gates, unbounded fan-in AND-gates and OR-gates, and f_i-oracle gates ($1 \leq i \leq k$). Here, an f_i-oracle gate receives an ordered tuple of inputs $x_1, x_2, \ldots, x_n$ and outputs the bits of $f_i(x_1 x_2 \cdots x_n)$. By taking the characteristic function of a language, we can also allow a language $L_i \subseteq \{0,1\}^*$ in place of f_i. Note that the function class $\mathsf{AC}^0(f_1, \ldots, f_k)$ is closed under composition (since the composition of two AC^0-circuits is again an AC^0-circuit). We write $\mathsf{AC}^0(\mathsf{NL}, f_1, \ldots, f_k)$ for $\mathsf{AC}^0(\mathsf{GAP}, f_1, \ldots, f_k)$ where GAP is the NL-complete graph accessibility problem. The class $\mathsf{AC}^0(\mathsf{NL})$ is studied in [9]. It has several alternative characterizations and can be viewed as a nondeterministic version of functional logspace. As remarked in [9], the restriction of $\mathsf{AC}^0(\mathsf{NL})$ to 0-1 functions is NL. Clearly, every logspace-computable function belongs to $\mathsf{AC}^0(\mathsf{NL})$: the NL-oracle can be used to directly compute the output bits of a logspace-computable function.

2.2 Counting complexity classes

Let Σ be a finite alphabet. The counting class $\#\mathsf{L}$ consists of all functions $f : \Sigma^* \to \mathbb{N}$ for which there is a logarithmic space bounded nondeterministic Turing machine M such that for every $w \in \Sigma^*$, $f(w)$ is the number of accepting computation paths of M on input w. The class $\mathsf{C}_=\mathsf{L}$ contains all languages A for which there are two functions $f_1, f_2 \in \#\mathsf{L}$ such that for every $w \in \Sigma^*$, $w \in A$ if and only if $f_1(w) = f_2(w)$. The class $\mathsf{C}_=\mathsf{L}$ is closed under logspace many-one reductions. One canonical $\mathsf{C}_=\mathsf{L}$-complete problem is the following: the input consists of two dags G_1 and G_2 and vertices s_1, t_1 (in G_1) and s_2, t_2 (in G_2), and it is asked whether the number of different paths from s_1 to t_1 in G_1 is equal to the number of different paths from s_2 to t_2 in G_2. An important $\mathsf{C}_=\mathsf{L}$-complete problem is the question whether the determinant of a given integer matrix is zero [83, 86].

An *NAuxPDA* is a nondeterministic Turing machine with an additional pushdown store. The class $\mathsf{LogCFL} \subseteq \mathsf{NC}^2$ is the class of all languages that can be accepted by a polynomial time bounded NAuxPDA whose work tape is logarithmically bounded (but the pushdown store is unbounded). If we assign to the input the number of accepting computation paths of such an NAuxPDA, we obtain the counting class $\#\mathsf{LogCFL}$. The class $\mathsf{C}_=\mathsf{LogCFL}$ contains all languages A for which there are two functions $f_1, f_2 \in \#\mathsf{LogCFL}$ such that for every $w \in \Sigma^*$, $w \in A$ if and only if $f_1(w) = f_2(w)$.

Let $\mathsf{DET} = \mathsf{AC}^0(\mathsf{det})$ where det is the function that maps a binary encoded integer matrix to the binary encoding of its determinant, see [28]. Actually, Cook defined DET as $\mathsf{NC}^1(\mathsf{det})$ [28], but the above definition via AC^0-circuits seems to be more natural. For instance, it implies that DET is equal to the $\#\mathsf{L}$-hierarchy, see also the discussion in [30].

We defined DET as a function class, but the definition can be extended to languages by considering their characteristic functions. It is well known that $\mathsf{NL} \subseteq \mathsf{DET} \subseteq \mathsf{NC}^2$, see e.g. [10]. From $\mathsf{NL} \subseteq \mathsf{DET}$, it follows easily that $\mathsf{AC}^0(\mathsf{NL}, f_1, \ldots, f_k) \subseteq \mathsf{DET}$ whenever $f_1, \ldots, f_k \in \mathsf{DET}$.

2.3 Randomized complexity classes

The class RP is the set of all problems A for which there exists a polynomial time bounded randomized Turing machine R such that: (i) if $x \in A$ then R accepts x with probability at least $1/2$, and (ii) if $x \notin A$ then R accepts x with probability 0. The class coRP is the class of all complements of problems from RP.

To define a randomized version of NC^i, one uses circuit families with additional inputs. So, let the n-th circuit $\mathcal{C}_n$ in the family have n normal input gates plus m random input gates where m

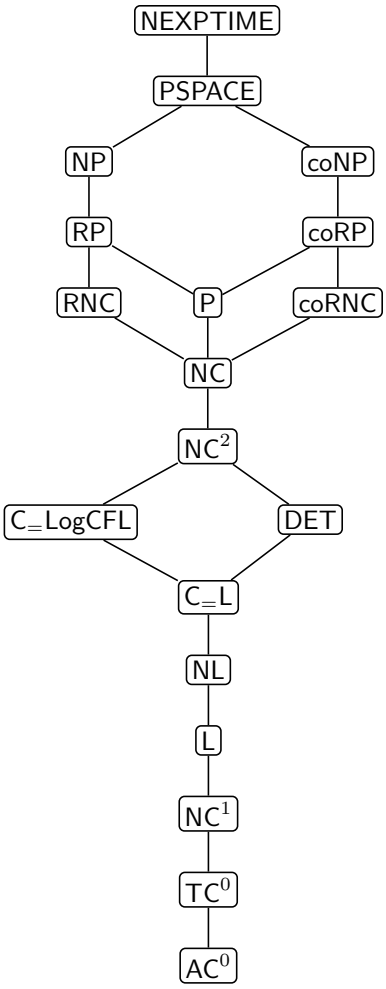


Figure 2.1: Illustration of the inclusions of the considered complexity classes for decision problems.

is polynomially bounded in n. For an input $x \in \{0,1\}^n$ one defines the acceptance probability as

$$\mathsf{Prob}[\mathcal{C}_n \text{ accepts } x] = \frac{|\{y \in \{0,1\}^m \mid \mathcal{C}_n(x,y) = 1\}|}{2^m}$$

Here, $\mathcal{C}_n(x,y) = 1$ means that the circuit $\mathcal{C}_n$ evaluates to 1 if the i-th normal input gate gets the i-th bit of the input string x, and the i-th random input gate gets the i-th bit of the random string y. Then, the class RNC^i is the class of all problems A for which there exists a polynomial size circuit family $(\mathcal{C}_n)_{n\geq 0}$ of depth $O(\log^i n)$ with random input gates that uses NOT-gates and AND-gates and OR-gates of fan-in two such that for all inputs $x \in \{0,1\}^*$ of length n: (i) if $x \in A$, then $\mathsf{Prob}[\mathcal{C}_n \text{ accepts } x] \geq 1/2$, and (ii) if $x \notin A$, then $\mathsf{Prob}[\mathcal{C}_n \text{ accepts } x] = 0$. As usual, coRNC^i is the class of all complements of problems from RNC^i. Section B.9 in [40] contains several problems that are known to be in RNC, but which are not known to be in NC; the most prominent example is the existence of a perfect matching in a graph. The diagram in Figure 2.1 illustrates the inclusion properties of the mentioned complexity classes for decision problems.

Chapter 3

Algebraic structures

3.1 General algebraic structures

An *algebraic structure* $\mathcal{A} = (D, f_1, \ldots, f_k)$ consists of a non-empty *domain* D and operations $f_i : D^{n_i} \to D$ for some $n_i \in \mathbb{N}$ and $1 \leq i \leq k$. We often identify the domain with the structure, if it is clear from the context. A *substructure* of $\mathcal{A}$ is a subset $B \subseteq D$ that is closed under each of the operations f_i. We identify B with the structure $(B, g_1, \ldots, g_k)$ where $g_i : B^{n_i} \to B$ is the restriction of f_i to B^{n_i} for all $1 \leq i \leq k$. In this work we only consider finitely generated (f.g.) structures, i.e., structures $\mathcal{A}$ where there is a finite set $I \subseteq D$ such that every $a \in D$ can be generated by iterative application of functions f_i on elements of I.

We mainly deal with f.g. semigroups and semirings. In the following we present the necessary background about these structures. For further details on semigroup theory (resp., semiring theory) see [74] (resp., [39]).

3.2 (Semi-)groups and monoids

Definition 3.1 (semigroup, monoid, group)**.**

- *A semigroup $\mathcal{S} = (S, \cdot)$ is a non-empty set with an associative operation $\cdot : S \times S \to S$.*
- *If there exists an identity element $e \in S$ in a semigroup S, i.e., $e \cdot s = s \cdot e = s$ for all $s \in S$, then S is a monoid. We usually denote the identity element of a monoid with 1.*
- *A group G is a monoid where for every $s \in G$ there is a $t \in G$ such that $s \cdot t = t \cdot s = 1$. In this case t is unique and denoted by s^{-1}.*

In the following we write st for $s \cdot t$. With S^1 we denote the monoid that is obtained from the semigroup S by adding a fresh element 1, which becomes the identity element of S^1. Thus, we extend the multiplication to $S^1 = S \cup \{1\}$ by setting $s1 = 1s = s$ for all $s \in S \cup \{1\}$.

Definition 3.2 (commutative, abelian, commutator)**.** *Let S be a semigroup. If $st = ts$ for all $s, t \in S$, we call S commutative. A commutative group is called* abelian *group. For elements s, t in a group G we denote with $[s, t] = s^{-1}t^{-1}st$ the commutator of s and t.*

Definition 3.3 (idempotent)**.** *Let S be a semigroup. An element $s \in S$ is called* idempotent *if $ss = s$. The set of all idempotents of S is denoted by $E(S)$ or simply E in the case that the semigroup S is clear from the context.*

It is well-known that for every finite semigroup S and every $s \in S$ there exists an $n \geq 1$ such that s^n is idempotent. By taking the smallest common multiple of all these n, one obtains an $\omega \geq 1$ such that s^ω is idempotent for all $s \in S$. In particular, every finite semigroup contains at least one idempotent element.

Lemma 3.4. *Every monoid that is not a group contains at least two distinct idempotents.*

Proof. Let M be a monoid and $\omega \in \mathbb{N}$ such that s^ω is idempotent for all $s \in S$. Of course, the identity 1 is idempotent. Assume that 1 is the unique idempotent in M. Then for all $s \in M$ $s^\omega = 1$, wich implies that $s^{-1} = s^{\omega-1}$ and M is a group. □

Definition 3.5 (direct product)**.** *For two semigroups $(S, \cdot_S)$ and $(T, \cdot_T)$ the direct product of S and T is the semigroup $(S \times T, \cdot)$ where $S \times T = \{(s,t) \mid s \in S, t \in T\}$ with the operation $(s_1, t_1) \cdot (s_2, t_2) = (s_1 \cdot_S s_2, t_1 \cdot_T t_2)$. For the n-fold direct product $S \times S \cdots \times S$ we write S^n.*

By the fundamental theorem of finitely generated abelian groups we know that every finitely generated abelian group is a finite direct product of copies of $\mathbb{Z}$ and $\mathbb{Z}_n$ [76].

Definition 3.6 (subsemigroup, submonoid, subgroup)**.** *A subsemigroup (resp.,submonoid; subgroup) of a semigroup $(S, \cdot)$ is a subset $T \subseteq S$ such that $(T, \cdot)$ is a semigroup (resp., monoid; group).*

Note that in case S is a monoid and T is a submonoid of S, we do not require that the identity element of $(T, \cdot)$ is 1 (the identity element of S). But, clearly, the identity element of the submonoid T must be an idempotent element of S. In fact, for every idempotent $e \in E(S)$, the set $eSe = \{ese \mid s \in S\}$ is a submonoid of S with identity e. The submonoid eSe is the maximal submonoid of S whose identity element is e.

Definition 3.7 (ideal)**.** *For a semigroup S a subset $I \subseteq S$ is called semigroup ideal if for all $s \in S, a \in I$ we have $sa, as \in I$.*

A *minimal* ideal is a non-empty ideal that contains no other non-empty ideal. If S is a finite semigroup, then $SES = S^n$ where $n = |S|$. Moreover, $S^n = S^m$ for all $m \geq n$.

Definition 3.8 (trivial, aperiodic)**.** *A group G is trivial, if $|G|$=1. A semigroup S is* aperiodic *if every subgroup of S is trivial.*

Definition 3.9 (homomorphism, embedding, isomorphism)**.** *For two semigroups $(S_1, \cdot_1), (S_2, \cdot_2)$ a mapping $f : S_1 \to S_2$ is a homomorphism if for all $s, t \in S_1$ $f(s \cdot_1 t) = f(s) \cdot_2 f(t)$. S_1 embeds into S_2 if there is an injective homomorphism $f : S_1 \to S_2$. In this case we call f an embedding. A bijective homomorphism is called isomorphism. If there is an isomorphism $f : S_1 \to S_2$, then S_1 and S_2 are called isomorphic. We denote this by $S_1 \cong S_2$.*

For a semigroup (resp. group) S and a non-empty subset $T \subseteq S$ we denote by $\langle T \rangle$ the subsemigroup of S that is generated by T. It consists of all finite non-empty products of elements from T (resp. $T \cup T^{-1}$).

Definition 3.10 (cyclic)**.** *Let G be a group. If there is a singleton t such that G is generated by $\{t\}$, then G is called cyclic.*

Definition 3.11 (free monoid)**.** *For a set Σ, the free monoid generated by Σ is the set Σ^* of all finite words over Σ together with the operation of concatenation.*

Definition 3.12 (free group)**.** *The free group of rank k, denoted by F_k, is the group that is generated by the set $\{a_1, \ldots, a_k\}$ such that two elements are different unless they are equal by the group axiom $a_i a_i^{-1} = 1$.*

Definition 3.13 (coset, normal subgroup, quotient)**.** *For a group G and a subgroup U of G the left (resp. right) cosets of U are the sets $aU = \{a \cdot b \mid b \in U\}$ (resp. $Ua = \{b \cdot a \mid b \in U\}$) for $a \in G$. If the right and left cosets of U coincide, U is a normal subgroup of G and the set of cosets of U is called quotient of G and is denoted by G/U. It is a group with the operation $aU \cdot bU = abU$. A normal subgroup U is called of finite index if G/U is finite.*

Definition 3.14 (commutator subgroup)**.** *Let G be a group. The commutator subgroup of G is the group $[G, G] := \langle \{[s,t] \mid s, t \in G\} \rangle$.*

The commutator subgroup $[G, G]$ of a group G is the smallest normal subgroup of G such that the quotient is abelian (i.e., if U is a normal subgroup of G and G/U is abelian, then $[G, G]$ is a subgroup of U) [76].

Definition 3.15 (presentation, finitely presented)**.** *The group G has the presentation $\langle a_1, \ldots, a_k \mid R\rangle$ with $R \subseteq F_k$ if it is isomorphic to $F_k/\langle R\rangle$. G is finitely presented if there is a finite presentation of G.*

Definition 3.16 (solvable, (strongly) polycyclic)**.**

- *A (n-step) solvable group G is a group which has a subnormal series $G = G_n \rhd G_{n-1} \rhd G_{n-2} \rhd \cdots \rhd G_1 \rhd G_0 = 1$ (i.e., G_i is a normal subgroup of G_{i+1} for all $0 \leq i \leq n-1$) such that every quotient G_{i+1}/G_i is abelian ($0 \leq i \leq n-1$).*
- *A semigroup S is solvable if every subgroup of S is a solvable group.*
- *If G is a solvable group and every quotient G_{i+1}/G_i is cyclic, then G is called polycyclic.*
- *If G is a solvable group and $G_{i+1}/G_i \cong \mathbb{Z}$ for all $0 \leq i \leq n-1$, then G is called strongly polycyclic.*

The number of $0 \leq i \leq n-1$ such that $G_{i+1}/G_i \cong \mathbb{Z}$ is called the *Hirsch length* of G; it does not depend on the chosen subnormal series.

Since abelian groups are solvable, every commutative semigroup is solvable. A group is polycyclic if and only if it is solvable and every abelian subgroup is finitely generated [50].

Definition 3.17 (lower central series, nilpotent group)**.** *For a group G its* lower central series *is the series $G = G_1 \rhd G_2 \rhd G_3 \rhd \cdots$ of subgroups where $G_{i+1} = [G_i, G]$ which is the subgroup generated by all commutators $[g, h]$ with $g \in G_i$ and $h \in G$. Indeed, G_{i+1} is a normal subgroup of G_i. The group G is nilpotent if its lower central series terminates after finitely many steps in the trivial group 1.*

Definition 3.18 (virtually abelian, virtually abelian, metabelian)**.** *A group G is*

- *virtually abelian if there is a normal abelian subgroup U of G of finite index.*
- *virtually nilpotent if there is a normal nilpotent subgroup U of G of finite index.*
- *called metabelian if the commutator subgroup $[G, G]$ is abelian. In other words, the metabelian groups are the 2-step solvable groups.*

Notice that even if G is f.g. metabelian, this does not imply that G is polycyclic, since $[G, G]$ is not necessarily finitely generated.

Definition 3.19 (torsion-group)**.** *A group G is a* torsion-group *if for every $a \in G$ there is an $n \in \mathbb{N}$ such that $a^n = 1$. If there is no element in G with that property, then we call G* torsion-free*.*

We need the following results about nilpotent and solvable groups:

Theorem 3.20 ([76, Chapter 5])**.** *Every subgroup and every quotient of a solvable (resp., nilpotent) group G is solvable (resp., nilpotent) again.*

Theorem 3.21 ([50, Theorem 17.2.2])**.** *Every f.g. nilpotent group G has a torsion-free normal subgroup H of finite index (which is also f.g. nilpotent).*

3.3 (Semi-)rings and fields

Definition 3.22 (semiring)**.** *A* semiring $\mathcal{R} = (R, +, \cdot)$ *consists of a non-empty set* R *with two operations* $+$ *and* $\cdot$ *such that*

- $(R, +)$ *is a commutative semigroup,*
- $(R, \cdot)$ *is a semigroup and*
- $\cdot$ *left- and right-distributes over* $+$*, i.e.,* $a \cdot (b + c) = ab + ac$ *and* $(b + c) \cdot a = ba + ca$*.*

Note that we neither require the existence of an additive identity 0 nor the existence of a multiplicative identity 1. We denote with $\mathcal{R}^+ = (R, +)$ the additive semigroup of R and with $\mathcal{R}^\bullet = (R, \cdot)$ the multiplicative semigroup of R. If $(R, \cdot)$ is commutative, we call R commutative. For $n \geq 1$ and $r \in R$ we write $n \cdot r$ or just nr for $r + \cdots + r$ where r is added n times.

With R^0 we denote the semiring that is obtained from the semiring R by adding a fresh element 0, which becomes the identity element of $(R^0)^+$ and is absorbing w.r.t. the multiplication. Thus, we extend the multiplication and addition to $R^0 = R \cup \{0\}$ by setting $a0 = 0a = 0$ and $0 + a = a$ for all $a \in R \cup \{0\}$.

Definition 3.23 (unitary, ring, $\{0,1\}$-free)**.** *A semiring* R *is*

- *unitary if* $(R, \cdot)$ *is a monoid.*
- *a ring if* $(R, +)$ *is a group.*
- $\{0,1\}$*-free if* R *does not contain a subsemiring* T *with an additive identity* 0 *and a multiplicative identity* $1 \neq 0$*.*

Definition 3.24 (homomorphism, isomorphism, ideal)**.** *For two semirings* $(R_1, +_1, \cdot_1), (R_2, +_2, \cdot_2)$ *a mapping* $f : S_1 \to S_2$ *is a (ring-)homomorphism (resp., isomorphism) if it is a homomorphism (resp. isomorphism) for the additive and multiplicative semigroups of* R_1 *and* R_2*. For a semiring* $(R, +, \cdot)$ *a non-empty subset* $I \subseteq R$ *is an ideal, if it is closed under addition and an ideal for the multiplicative semigroup.*

For a unitary semiring R we identify a natural number $n \geq 1$ with the n-fold sum of 1, i.e., $n = n \cdot 1$. Note that in the last definition for the semiring T we do not require that $a \cdot 0 = 0 \cdot a = 0$ for all $a \in T$. The class of $\{0,1\}$-free finite semirings has several characterizations:

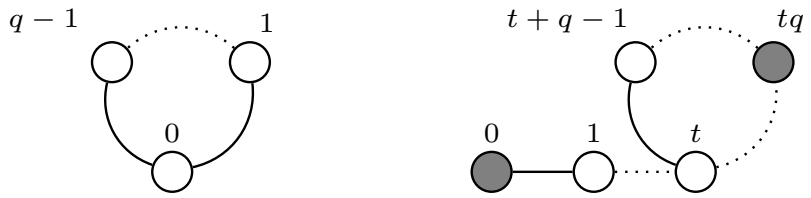
Lemma 3.25. *For a finite semiring* R*, the following are equivalent:*

1. R *is not* $\{0,1\}$*-free.*
2. R *contains the Boolean semiring* $\mathbb{B}_2$ *or the ring* $\mathbb{Z}_q$ *for some* $q \geq 2$ *as a subsemiring.*
3. R *is divided by* $\mathbb{B}_2$ *or* $\mathbb{Z}_q$ *for some* $q \geq 2$ *(i.e.,* $\mathbb{B}_2$ *or* $\mathbb{Z}_q$ *is the image of a homomorphism in a subsemiring of* R*).*
4. *There exist elements* $0, 1 \in R$ *such that* $0 \neq 1$*,* $0 + 0 = 0$*,* $0 + 1 = 1$*,* $0 \cdot 1 = 1 \cdot 0 = 0 \cdot 0 = 0$*, and* $1 \cdot 1 = 1$ *(but* $1 + 1 \neq 1$ *is possible).*

Proof. $(1 \Rightarrow 2)$: Let T be a subsemiring of R which has a zero element 0 and a one element $1 \neq 0$. Note that $0 \cdot 0 = 0 \cdot 0 + 0 = 0 \cdot 0 + 1 \cdot 0 = (0 + 1) \cdot 0 = 1 \cdot 0 = 0$. Let $T' = \{0\} \cup \{k \cdot 1 \mid k \in \mathbb{N}\}$, which is the subsemiring generated by 0 and 1. It is isomorphic to some semiring $B(t, q)$ ($t \geq 0$, $q \geq 1$), which is the semiring $(\mathbb{N}, +, \cdot)$ modulo the congruence relation $\theta_{t,q}$ defined by

$$i \,\theta_{t,q}\, j \iff i = j \text{ or } [i, j \geq t \text{ and } i \equiv j \pmod{q}].$$

Since $0 \neq 1$, we have $(t, q) \neq (0, 1)$. If $t = 0$, then $B(0, q)$ is isomorphic to $\mathbb{Z}_q$ for $q \geq 2$. If $t \geq 1$, then choose $a \geq t$ such that q divides a, for example $a = qt$. Then $\{0, a \cdot 1\}$ is a subsemiring isomorphic to the Boolean semiring $\mathbb{B}_2$. (See Figure 3.1 for an illustration.)



(a) $\mathbb{Z}_q$ is embedded (b) $\mathbb{B}_2$ is embedded

Figure 3.1: Semirings generated by 0 and 1.

$(2 \Rightarrow 3)$: This implication is trivial.

$(3 \Rightarrow 4)$: Assume that $\varphi : T \to T'$ is a homomorphism from a subsemiring T of R to T' where the latter is $\mathbb{B}_2$ or $\mathbb{Z}_q$ with $q \geq 2$. Let $n \geq 1$ be such that $n \cdot x$ is additively idempotent and x^n is multiplicatively idempotent for all $x \in R$. Then $n \cdot x^n$ is is additively and multiplicatively idempotent for all $x \in R$. Let $a, e \in T$ be such that $\varphi(a) = 0$ and $\varphi(e) = 1$. We can replace a by $n \cdot a^n$ and e by e^n. Then, $a + a = aa = a$ and $ee = e$. For $a' = n \cdot (eae)^n$ we have $\varphi(a') = 0$ and $a'e = ea' = a' + a' = a'a' = a'$. For $e' = a' + e$ we have $\varphi(e') = 1$ (hence, $a' \neq e'$) and $e'e' = a'a' + a'e + ea' + ee = a' + e = e'$, $a' + e' = a' + a' + e = e'$. Furthermore, we have $a'e' = a'(a' + e) = a' + a'e = a'$ and similarly $e'a' = a'$. Hence, a' and e' satisfy all equations from point 4.

$(4 \Rightarrow 1)$: Assume that there exist elements $0, 1 \in R$ such that $0 \neq 1$, $0 + 0 = 0$, $0 + 1 = 1$, $0 \cdot 1 = 1 \cdot 0 = 0 \cdot 0 = 0$, and $1 \cdot 1 = 1$. Consider the subsemiring generated by $\{0, 1\}$, which is $\{0\} \cup \{n \cdot 1 \mid n \geq 1\}$. By the above identities 0 (resp., 1) is an additive (resp., multiplicative) identity in this subsemiring. □

As a consequence of Lemma 3.25 (point 4), one can check in time $O(n^2)$ for a semiring of size n whether it is $\{0, 1\}$-free. We will not need this fact, since in our setting the finite semirings will be always fixed, i.e., not part of the input. Moreover, the class of all $\{0, 1\}$-free semirings is closed under taking subsemirings (this is trivial) and taking homomorphic images (by point 3). Finally, the class of $\{0, 1\}$-free semirings is also closed under direct products. To see this, assume that $R \times R'$ is not $\{0, 1\}$-free. Hence, there exists a subsemiring T of $R \times R'$ with an additive zero $(0, 0')$ and a multiplicative one $(1, 1') \neq (0, 0')$. W.l.o.g. assume that $0 \neq 1$. Then the projection $\pi_1(T)$ onto the first component is a subsemiring of R, where 0 is an additive identity and $1 \neq 0$ is a multiplicative identity. By these remarks, the class of $\{0, 1\}$-free finite semirings forms a pseudo-variety of finite semirings. Again, this fact will not be used, but it might be of independent interest. (For a background about pseudo-varieties consider e.g. [36].)

Figure 3.1 illustrates the possible semirings generated by the elements 0 and 1. In picture (b) the two elements that form the Boolean semiring are marked gray.

For a unitary ring R we denote by $R[x_1, \ldots, x_n]$ the *polynomial ring* over R which consists of all polynomials with coefficients from R and variables $x_1, \ldots, x_n$. Polynomials that contain more than one variable are called multivariate. Polynomials in $R[x]$ are called univariate.

Definition 3.26 (degree)**.** *For a univariate polynomial $p(x)$ let $\deg(p)$ be the degree of p. It is the largest number d such that x^d appears in a monomial of p. For a multivariate polynomial $p(x_1, \ldots, x_k) \in R[x_1, \ldots, x_k]$ let $\deg(p, x_i) := \deg(p(1, \ldots, 1, x_i, 1, \ldots, 1))$ be the degree of p in the variable x_i.*

With $E(R)$ we denote the set of multiplicative idempotents of R, i.e., those $e \in R$ with $e^2 = e$. Note that for every multiplicative idempotent $e \in E(R)$, eRe is a unitary subsemiring of R.

Definition 3.27 (free semiring)**.** *For a given non-empty set Σ, the free semiring $\mathbb{N}[\Sigma]$ generated by Σ consists of all mappings $f : \Sigma^+ \to \mathbb{N}$ such that the support of f defined by $\mathsf{supp}(f) := \{w \in \Sigma^+ \mid f(w) \neq 0\}$ is finite and non-empty. Addition is defined pointwise,*

i.e., $(f+g)(w) = f(w) + g(w)$*, and multiplication is defined by the convolution:* $(f \cdot g)(w) = \sum_{w=uv} f(u) \cdot g(v)$ *where the sum is taken over all factorizations* $w = uv$ *with* $u, v \in \Sigma^+$*.*

We treat an element $f \in \mathbb{N}[\Sigma]$ as a non-commutative polynomial $\sum_{w \in \mathsf{supp}(f)} f(w) \cdot w$. Then addition (resp. multiplication) in $\mathbb{N}[\Sigma]$ corresponds to addition (resp. multiplication) of non-commutative polynomials. Words $w \in \mathsf{supp}(f)$ are also called *monomials* of f. A word $w \in \Sigma^+$ is identified with the non-commutative polynomial $1 \cdot w$, i.e., the mapping f with $\mathsf{supp}(f) = \{w\}$ and $f(w) = 1$. For every semiring R which is generated by Σ there exists a canonical surjective homomorphism from $\mathbb{N}[\Sigma]$ to R which evaluates non-commutative polynomials over Σ. Since a semiring is not assumed to have a multiplicative identity (resp., additive identity), we have to exclude the empty word from $\mathsf{supp}(f)$ for every $f \in \mathbb{N}[\Sigma]$ (resp., exclude the mapping f with $\mathsf{supp}(f) = \emptyset$ from $\mathbb{N}[\Sigma]$).

Definition 3.28 (field, subfield)**.**

- *A* field $(F, +, \cdot)$ *is a ring where* $(F \setminus \{0\}, \cdot)$ *is an abelian group.*
- *A subfield of* $(F, +, \cdot)$ *is a subset* $K \subseteq F$ *such that* $(K, +, \cdot)$ *is also a field.*

If K is a subfield of a field F, we call F an *extension field* of K. To make clear that F is an extension of K we will denote the extension by $[F : K]$. The smallest subfield of a field F is called *prime field*. It is always isomorphic to the field of rational numbers or to a finite field $\mathbb{F}_p$ for some prime p. For a field K, an extension $[F : K]$ of K and a subset $S \subseteq F$, we define by $K(S)$ the smallest subfield of $[F : K]$ which contains K and S. For a single element $s \in [F : K]$, instead of $K(\{s\})$ we write $K(s)$. In this case we call $K(s)$ a *simple extension* and s a *primitive element* of the extension. A field-extension $[F : K]$ can be considered as a vector space where the scalars are elements of K. The degree of this vector space is called *degree of the extension*. An extension of finite degree is called *finite extension*. For a field F and a subfield K we call $a \in F$ *algebraic over* K, if there is a non-zero polynomial $g(x)$ with coefficients in K such that $g(a) = 0$. If there is no other such polynomial with lesser degree, we call g minimal polynomial of a over F. If every $a \in F$ is algebraic over K we call $[F : K]$ an *algebraic extension* of K.

Definition 3.29 (algebraic closure, separable extension)**.**

- *A field* F *is* algebraically closed*, if* F *contains a root for every non-constant polynomial in* $F[x]$*. An* algebraic closure *of a field* F *is an algebraic extension of* F *that is algebraically closed.*
- *A polynomial* $g(x)$ *over a field* F *is called* separable*, if its roots are distinct in an algebraic closure of* F*. An extension* $[F : K]$ *is called* separable extension*, if it is an algebraic extension and for every* $a \in F$ *the minimal polynomial of* a *over* K *is separable.*

Theorem 3.30 (primitive element theorem [92])**.** *For every finite separable extension* $[F : K]$ *there is some* a *in* F *such that* $[F : K] = K(a)$*.*

Notice that in this case a is algebraic over K, since $[F : K]$ is an algebraic extension.

Definition 3.31 (characteristic)**.** *The characteristic of a field* F *is the smallest number* $d \geq 1$ *such that* $d \cdot 1 = 0$ *in* R*. If no such* d *exists we set* $d = 0$*.*

The characteristic of a field F is either a prime number or 0.

3.4 Matrix groups

In this thesis we are concerned with certain subclasses of matrix groups, which are defined in the following:

Definition 3.32 (linear group)**.** *A group* G *is linear if it is isomorphic to a subgroup of* $\mathsf{GL}_d(F)$ *(the group of all invertible* $(d \times d)$*-matrices over the field* F*) for some field* F*.*

Let A be a square matrix of dimension d over some commutative unitary ring R. With $A[i,j]$ we denote the entry of A in row i and column j. The matrix A is called *triangular* if $A[i,j] = 0$ whenever $i > j$, i.e., all entries below the main diagonal are 0. A *unitriangular matrix* is a triangular matrix A such that $A[i,i] = 1$ for all $1 \leq i \leq d$, i.e., all entries on the main diagonal are 1. The set of unitriangular matrices of dimension d over R is denoted by $\mathsf{UT}_d(R)$. It is well known that for every commutative unitary ring R, the set $\mathsf{UT}_d(R)$ is a group (with respect to matrix multiplication).

Definition 3.33 (unitriangular group)**.** *A group is unitriangular if it is isomorphic to* $\mathsf{UT}_d(R)$ *for some commutative unitary ring* R*.*

Let $1 \leq i, j \leq d$. With $T_{i,j}$ we denote the matrix such that all entries in the main diagonal are 1, $T_{i,j}[i,j] = 1$ and all other entries are 0. The notation $T_{i,j}$ does not specify the dimension d of the matrix, but the dimension will be always clear from the context. The group $\mathsf{UT}_d(\mathbb{Z})$ is generated by the finite set $\Gamma_d = \{T_{i,i+1} \mid 1 \leq i < d\}$, see e.g. [22].

We need the following result:

Theorem 3.34 ([50, Theorem 17.2.5])**.** *For every torsion-free f.g nilpotent group* G *there exists* $d \geq 1$ *such that* G *can be embedded into* $\mathsf{UT}_d(\mathbb{Z})$*.*

Together with Theorem 3.21 this leads immediately to the following result:

Corollary 3.35. *Every f.g. nilpotent group* G *has a normal subgroup* H *that can be embedded into* $\mathsf{UT}_d(\mathbb{Z})$ *for some* $d \in \mathbb{N}$*.*

We will make use of the following lemma, which shows how to encode multiplication of integers by unitriangular matrices. See [58] for a proof.

Lemma 3.36. *For all* $a, b \in \mathbb{Z}$ *and* $1 \leq i < j < k \leq d$ *we have* $[T^a_{i,j}, T^b_{j,k}] = T^{ab}_{i,k}$*.*

The *special linear group* $\mathsf{SL}_d(\mathbb{Z})$ is the group of $d \times d$-matrices over $\mathbb{Z}$ with determinant equal to 1. In $\mathsf{SL}_3(\mathbb{Z})$ the multiplication of integers can be encoded in the following way, where a proof can be found for instance in [57].

Lemma 3.37. *Let* $T_{i,j}$ *for* $i, j \in \{1,2,3\}$ *and* $i \neq j$ *be as defined above. Then for all* $a, b \in \mathbb{Z}$ *and* $k \in \{1,2,3\} \setminus \{i,j\}$ *we have* $[T^a_{k,j}, T^{-b}_{i,k}] = T^{ab}_{i,j}$*.*

Auslander and Swan [16, 81] proved that the polycyclic groups are exactly the solvable groups of integer matrices, so every polycyclic group is linear.

3.5 Wreath products

Definition 3.38. *Let* G *and* H *be groups. The restricted wreath product* $H \wr G$ *is defined as follows:*

- *Elements of* $H \wr G$ *are pairs* (f, g) *where* $g \in G$ *and* $f : G \to H$ *is a mapping such that* $f(a) \neq 1_H$ *for only finitely many* $a \in G$ *(*1_H *is the identity element of* H*).*
- *The multiplication in* $H \wr G$ *is defined as follows: let* $(f_1, g_1), (f_2, g_2) \in H \wr G$*. Then* $(f_1, g_1)(f_2, g_2) = (f, g_1 g_2)$ *where* $f(a) = f_1(a) f_2(g_1^{-1} a)$*.*

For readers, who have not seen this definition before, the following intuition might be helpful: an element $(f, g) \in H \wr G$ can be seen as a finite collection of elements of H that are sitting in certain elements of G (the mapping f) together with a distinguished element of G (the element g), which can be seen as a cursor moving around G. If we want to compute the product $(f_1, g_1)(f_2, g_2)$, we do this as follows: first, we shift the finite collection of H-elements that corresponds to the mapping f_2 by g_1: if the element $h \in H \setminus \{1_H\}$ is sitting in $a \in G$ (i.e., $f(a) = h$), then we remove h from a and put it to the new location $g_1 a \in G$. This new collection corresponds to the mapping

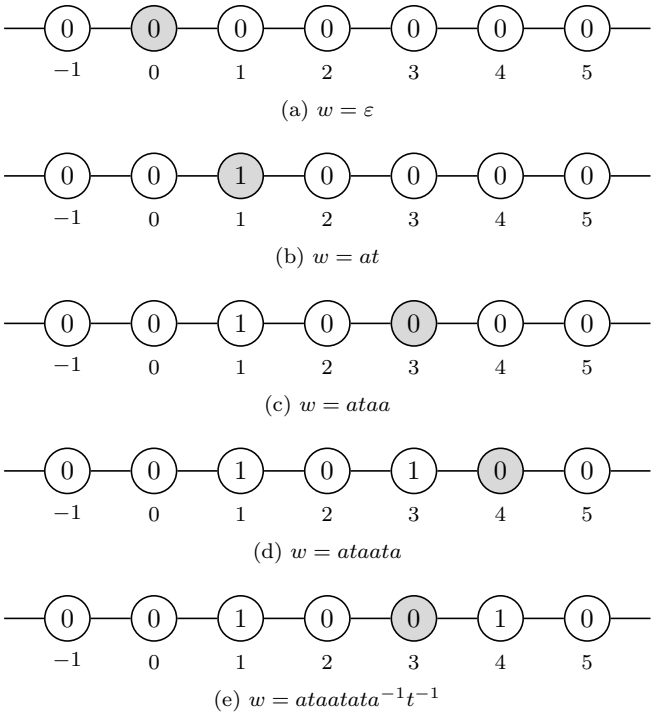


Figure 3.2: The element in $\mathbb{Z} \wr \mathbb{Z}$ corresponding to $w = ataatata^{-1}t^{-1}$.

$f_2' : a \mapsto f_2(g_1^{-1}a)$. After this shift, we multiply the two collections of H-elements pointwise: if in $a \in G$ the elements h_1 and h_2 are sitting (i.e., $f_1(a) = h_1$ and $f_2'(a) = h_2$), then we put the product h_1h_2 into the G-location a. Finally, the new distinguished G-element (the new cursor position) becomes g_1g_2.

Example 3.39. *The wreath product $\mathbb{Z} \wr \mathbb{Z}$ is generated by $\{a, t, a^{-1}, t^{-1}\}$ and can be seen as the number line where on every position sits an integer (only on finite many positions not equal to zero) and a cursor is moving to the right (by a) or left (by a^{-1}) on the number line and increases (by t), resp., decreases (by t^{-1}) the integer sitting on the cursers position. At the beginning every integer is 0 and the curser is at position 0. In this setting the word $w = ataatata^{-1}t^{-1}$ evaluates to the element in Figure3.2, where the position of the curser is marked grey.*

Lemma 3.40. *The group $(A \times B) \wr G$ embeds into $(A \wr G) \times (B \wr G)$.*

Proof. Let $\pi_A : A \times B \to A$ be the natural projection morphism and similarly let $\pi_B : A \times B \to B$. We define an embedding $\varphi : (A \times B) \wr G \to (A \wr G) \times (B \wr G)$ by

$$\varphi(f, g) = \Big((\pi_A \circ f, g), (\pi_B \circ f, g) \Big).$$

Clearly, φ is injective. Moreover, it is easy to see that φ is a group homomorphism.

□

A proof of the following lemma can be found for instance in [56].

Lemma 3.41. *Let K be a subgroup of H of finite index m and let G be a group. Then $G^m \wr K$ is isomorphic to a subgroup of index m of $G \wr H$.*

Chapter 4

The classical word problem

4.1 Introduction

Before we consider the circuit evaluation problem, we first have a look at a more classical problem of computational group theory that can be seen as the origin of the circuit evaluation problem for groups, the so-called (classical) word problem. Already in 1911 Dehn stated three algebraic problems for finitely presented groups [31]: the word problem (given a word over the generators, does the word evaluate to the group identity?), the conjugacy problem (given two words w_1, w_2 over the generators, are they conjugate, i.e., is there an $x \in G$ such that $w_1 = xw_2x^{-1}$?) and the isomorphism problem (given two finite representations of groups, are the generated groups isomorphic?). Dehn himself found the first algorithms to solve the word problem for fundamental groups of orientable closed 2-dimensional manifolds [32]. But on the other hand he also suggested that it could be very hard for some groups to solve these problems. About 45 years later, after a formal concept of computability had been found, his suggestions were proved: Novikov [71] and Boone [23] independently showed that there are finitely presented groups with an undecidable word problem. But fortunately for many groups the word problem is decidable. Along the years many results about decidability and the complexity of the word problem for several groups were shown. For an extensive overview one might consult [12], [24], [25], [26], [37], [41], [49], [59], [73], resp. [77].

Formally the word problem for finitely generated groups can be stated as follows: let G be a finitely generated group and Σ the corresponding generating set. Then, as a monoid G is finitely generated by $\Sigma \cup \Sigma^{-1}$ (where $\Sigma^{-1} = \{a^{-1} \mid a \in \Sigma\}$ is a disjoint copy of Σ and a^{-1} stands for the inverse of the generator $a \in \Sigma$). The *word problem* for G is the following computational problem: given a string $w \in (\Sigma \cup \Sigma^{-1})^*$, does w evaluate to the identity of G.

In the following we consider the word problem for solvable linear groups. In this context it is interesting that Kharlampovich [42] proved that there exist 3-step solvable groups with an undecidable word problem. On the other hand, for every f.g. linear group Lipton and Zalcstein [53] and Simon [79] proved the following important result:

Theorem 4.1. *For every f.g. linear group the word problem can be solved in deterministic logarithmic space.*

Lipton and Zalcstein [53] proved this result for a linear group over a field of characteristic zero, whereas Simon [79] considered fields of prime characteristic. Theorem 4.1 implies that the word problem for every polycyclic group can be solved in logarithmic space. Robinson proved in his thesis that the word problem for a polycyclic group belongs to TC^0 [75], but his circuits are not uniform. For f.g. nilpotent groups, Robinson [75] proved that the word problem belongs to $\mathsf{DLOGTIME}$-uniform TC^0. Waack considered in [89] arbitrary f.g. solvable linear groups (which include the polycyclic groups) and proved that their word problems belong to logspace-uniform NC^1. In the next section, we combine Waack's technique with the famous division breakthrough

results by Hesse, Allender, and Barrington [43] to show that for every f.g. solvable linear group the word problem belongs to DLOGTIME-uniform TC^0. For the Baumslag-Solitar group $\mathsf{BS}_{1,2}$, which is solvable and linear, Diekert, Miasnikov and Weiß proved that also the conjugacy problem can be solved in DLOGTIME-uniform TC^0 [34], Miasnikov, Vassileva and Weiß proved that also the conjugacy problem for free solvable groups and wreath products of abelian groups is in DLOGTIME-uniform TC^0 [65], see also [91] for related results. The results of this chapter have appeared in [1].

4.2 The complexity of the classical word problem for finitely generated linear groups

We prove the aforementioned result on the classical word problem for f.g. solvable linear groups:

Theorem 4.2. *Let G be a f.g. linear group.*

- *If G is infinite solvable, then the word problem for G is complete for* DLOGTIME*-uniform* TC^0.
- *If G is virtually solvable (i.e., G has a solvable subgroup of finite index), then the word problem for G belongs to* DLOGTIME*-uniform* NC^1.

For the proof, we first have to consider the complexity of iterated multiplication and division with remainder for polynomials in $\mathbb{Z}[x_1, \ldots, x_k]$ given in standard representation (i.e., coefficients are given in binary encoding and exponents are given in unary encoding). Iterated multiplication of polynomials in the ring $\mathbb{Z}[x_1, \ldots, x_k]$ is the task of computing from a given list of polynomials $p_1, p_2, \ldots, p_n \in \mathbb{Z}[x_1, \ldots, x_k]$ the product polynomial $p_1 p_2 \cdots p_n$. Division with remainder in the ring $\mathbb{Z}[x]$ (later, we will generalize this to several variables) is the task of computing for given polynomials $s, t \in \mathbb{Z}[x]$ such that $t \neq 0$ and the leading coefficient of t is 1 the unique polynomials $s \bmod t$ and $s \operatorname{div} t$ such that $s = (s \operatorname{div} t) \cdot t + s \bmod t$ and $\mathsf{deg}(s \bmod t) < \mathsf{deg}(t)$.

We need generalizations of Theorem 2.1 to multivariate polynomials. In the following proofs we always use the fact that iterated addition, iterated multiplication and division with remainder of binary encoded integers can be done in DLOGTIME-uniform TC^0 [43].

Lemma 4.3. *Iterated multiplication of polynomials in the ring $\mathbb{Z}[x_1, \ldots, x_k]$ (resp., $\mathbb{F}_p[x_1, \ldots, x_k]$) belongs to* DLOGTIME*-uniform* TC^0.

Proof. We only prove the result for $\mathbb{Z}[x_1, \ldots, x_k]$; exactly the same proof also works for $\mathbb{F}_p[x_1, \ldots, x_k]$. For $d \geq 1$ let $\mathbb{Z}[x_1, \ldots, x_k]_d \subseteq \mathbb{Z}[x_1, \ldots, x_k]$ be the set of all polynomials $p \in \mathbb{Z}[x_1, \ldots, x_k]$ such that $\deg(p, x_i) \leq d$ for all $1 \leq i \leq k$. The mapping $\mathcal{U}_d : \mathbb{Z}[x_1, \ldots, x_k] \to \mathbb{Z}[z]$ is defined by

$$\mathcal{U}_d(p(x_1, x_2, \ldots, x_k)) = p(z^1, z^{d^1}, \ldots, z^{d^k}).$$

The mapping $\mathcal{U}_{d+1}$ restricted to $\mathbb{Z}[x_1, \ldots, x_k]_d$ is injective, since for a polynomial $p \in \mathbb{Z}[x_1, \ldots, x_k]_d$ we obtain the polynomial $\mathcal{U}_{d+1}(p)$ by replacing every monomial $a \cdot x_1^{n_1} \cdots x_k^{n_k}$ by the monomial $a \cdot z^N$ where N is the number with base-$(d+1)$ expansion $(n_1 \cdots n_k)$ (with the most significant digit on the right). Moreover, for all polynomials $p, q \in \mathbb{Z}[x_1, \ldots, x_k]$ and all $d \geq 2$ we have

$$\mathcal{U}_d(p+q) = \mathcal{U}_d(p) + \mathcal{U}_d(q) \text{ and } \mathcal{U}_d(pq) = \mathcal{U}_d(p)\mathcal{U}_d(q). \tag{4.1}$$

We can calculate $\mathcal{U}_d(p)$ for a given polynomial $p \in \mathbb{Z}[x_1, \ldots, x_k]$ and a given number $d \geq 2$ in unary representation, in DLOGTIME-uniform TC^0: for a monomial $a x_1^{n_1} \cdots x_k^{n_k}$ (which is represented by the tuple $(a, n_1, \ldots, n_k)$) we have to compute the pair $(a, \sum_{i=0}^{k-1} n_{i+1} d^i)$, which is possible in DLOGTIME-uniform TC^0. Similarly, we can compute $\mathcal{U}_{d+1}^{-1}(p)$ for a polynomial $p \in \mathcal{U}_{d+1}(\mathbb{Z}[x_1, \ldots, x_k]_d)$ in DLOGTIME-uniform TC^0: from a given monomial $a z^N$ (represented by the pair (a, N)) we have to compute the tuple $(a, n_1, \ldots, n_k)$ where $n_i = (N \operatorname{div} (d+1)^{i-1}) \bmod (d+1)$, which can be done in DLOGTIME-uniform TC^0.

We now multiply given polynomials $p_1, \ldots, p_n \in \mathbb{Z}[x_1, \ldots, x_k]$ in the following way, where all steps can be carried out in DLOGTIME-uniform TC^0 by the above remarks:

- Compute the number $d = \max\{\sum_{i=1}^{n} \deg(p_i, x_j) \mid 1 \leq j \leq k\}$. This number bounds the degree of the product polynomial $p_1 p_2 \cdots p_n$ in any of the variables $x_1, \ldots, x_n$, i.e., $p_1 p_2 \cdots p_n \in \mathbb{Z}[x_1, \ldots, x_k]_d$.
- Compute in parallel $s_i(z) := \mathcal{U}_{d+1}(p_i)$ for $1 \leq i \leq n$.
- Using Theorem 2.1, compute the product $S(z) = s_1(z)s_2(z)\cdots s_n(z)$, which is $\mathcal{U}_{d+1}(p_1 p_2 \cdots p_n)$ by (4.1).
- Finally, compute $\mathcal{U}_{d+1}^{-1}(S)$, which is $p_1 p_2 \cdots p_n$.

□

For polynomial division in several variables, we need a new distinguished variable. Therefore, we consider the polynomial ring $\mathbb{Z}[x_1, \ldots, x_k, y]$. We treat polynomials from this ring as polynomials in the variable y where coefficients are polynomials from $\mathbb{Z}[x_1, \ldots, x_k]$. We will only divide by a polynomial t for which the leading monomial $p(x_1, \ldots, x_n)y^m$ of t satisfies $p(x_1, \ldots, x_n) = 1$. This ensures that the coefficients of the quotient and remainder polynomial are again in $\mathbb{Z}[x_1, \ldots, x_k]$.

Lemma 4.4. *Division with remainder of polynomials in the ring $\mathbb{Z}[x_1, \ldots, x_k, y]$ (respectively, $\mathbb{F}_p[x_1, \ldots, x_k, y]$) belongs to* DLOGTIME*-uniform* TC^0.

Proof. Again, we only prove the result for $\mathbb{Z}[x_1, \ldots, x_k, y]$; exactly the same proof works for $\mathbb{F}_p[x_1, \ldots, x_k, y]$ as well. As in the proof of Lemma 4.3 consider the set $\mathbb{Z}[x_1, \ldots, x_k, y]_d \subseteq \mathbb{Z}[x_1, \ldots, x_k, y]$ of all polynomials in $\mathbb{Z}[x_1, \ldots, x_k, y]$ such that for every monomial $a \cdot x_1^{n_1} \cdots x_k^{n_k} y^n$ we have $n_1, \ldots, n_k, n < d$, and the mapping $\mathcal{U}_d : \mathbb{Z}[x_1, \ldots, x_k, y] \to \mathbb{Z}[z]$ with

$$\mathcal{U}_d(p(x_1, x_2, \ldots, x_k, y)) = p(z^1, z^{d^1}, \ldots, z^{d^{k-1}}, z^{d^k}).$$

Note that for polynomials $p, q \in \mathbb{Z}[x_1, \ldots, x_k, y]_d$ with $\deg(p, y) < \deg(q, y)$ we have $\deg(\mathcal{U}_{d+1}(p)) < \deg(\mathcal{U}_{d+1}(q))$, since the exponent of y becomes the most significant digit in the base-$(d+1)$ representation. Then, for all polynomials $s, t \in \mathbb{Z}[x_1, \ldots, x_k, y]_d$ (where the leading coefficient of t is 1) we have

$$\mathcal{U}_{d^2+1}(s \bmod t) = \mathcal{U}_{d^2+1}(s) \bmod \mathcal{U}_{d^2+1}(t).$$

To see this, assume that $s = qt + r$ with $\deg(r, y) < \deg(t, y)$ so that $r = s \bmod t$. We have $q, r \in \mathbb{Z}[x_1, \ldots, x_k, y]_{d^2}$, which can be checked by tracing the polynomial division algorithm. By (4.1) we have

$$\mathcal{U}_{d^2+1}(s) = \mathcal{U}_{d^2+1}(q)\mathcal{U}_{d^2+1}(t) + \mathcal{U}_{d^2+1}(r).$$

Moreover, $\deg(\mathcal{U}_{d^2+1}(r)) < \deg(\mathcal{U}_{d^2+1}(t))$. Hence

$$\mathcal{U}_{d^2+1}(r) = \mathcal{U}_{d^2+1}(s) \bmod \mathcal{U}_{d^2+1}(t).$$

Now we can compute the remainder $s \bmod t$ for given polynomials $s, t \in \mathbb{Z}[x_1, \ldots, x_k, y]$ (where the leading coefficient of t is 1) in DLOGTIME-uniform TC^0 as follows:

- compute the number $d = \max\{\deg(p, z) \mid p \in \{s, t\}, z \in \{x_1, \ldots, x_k, y\}\}$, so that $s, t \in \mathbb{Z}[x_1, \ldots, x_k, y]_d$.
- Compute in parallel $u(z) = \mathcal{U}_{d^2+1}(s)$ and $v(z) = \mathcal{U}_{d^2+1}(t)$.
- Compute, using Theorem 2.1, $R(z) = u(z) \bmod v(z)$, which is $\mathcal{U}_{d^2+1}(s \bmod t)$.
- Finally, compute $\mathcal{U}_{d^2+1}^{-1}(R)$ which is $s \bmod t$.

In the same way we can also compute the quotient, but we only will need the remainder $s \bmod t$ in the following. □

Finally, we will need the following result from [75]:

Theorem 4.5 ([75, Theorem 5.2])**.** *Let G be a f.g. group with a normal subgroup H of finite index. Then, the word problem for G is AC^0-reducible to the word problems for H and G/H.*

Now we are ready to prove Theorem 4.2.

Proof of Theorem 4.2. Let us first assume that G is f.g. solvable and linear over a field F. By a theorem of Mal'cev (see e.g. [90, Theorem 3.6]), G contains a normal subgroup H of finite index, which is triangularizable over a finite extension of F (i.e., H is isomorphic to a group of triangular matrices over a finite extension of F). Using Theorem 4.5 we know that the word problem for G is AC^0-reducible to the word problems for H and G/H. The latter is a finite solvable group, see Theorem 3.20. Hence, its word problem belongs to $\mathsf{DLOGTIME}$-uniform TC^0 by [18].

By the previous discussion, it suffices to show that the word problem for a f.g. triangular matrix group G over some field F belongs to $\mathsf{DLOGTIME}$-uniform TC^0. Let P be the prime field of F. We can replace F by the finite extension of P that is generated by all matrix entries in generators of G. It is known that the field extension $[F : P]$ has a separating transcendence base $\{x_1, \ldots, x_k\}$, which means that $[F : P(x_1, \ldots, x_k)]$ is a finite separable extension; see e.g. [92, Theorem 31].[1] Hence, the theorem of the primitive element applies, which says that F is generated over $P(x_1, \ldots, x_k)$ by a single element $\alpha \in F$, which is algebraic over $P(x_1, \ldots, x_k)$.

Assume now that $P = \mathbb{Q}$ (in case $P = \mathbb{F}_p$ for a prime p we have to replace in all arguments below $\mathbb{Z}$ (resp. $\mathbb{Q}$) by $\mathbb{F}_p$). Consider the minimal polynomial $p(y) \in \mathbb{Q}(x_1, \ldots, x_k)[y]$ of α. We can write it as

$$p(y) = y^m + \frac{p_1}{q} y^{m-1} + \frac{p_2}{q} y^{m-2} + \cdots + \frac{p_m}{q} \tag{4.2}$$

for some $m \in \mathbb{N}$ and $p_1, \ldots, p_m, q \in \mathbb{Z}[x_1, \ldots, x_k]$, $q \neq 0$. The element $\beta = \alpha \cdot q \in F$ also generates F over $P(x_1, \ldots, x_k)$, and its minimal polynomial is

$$r(y) = y^m + p_1 \cdot y^{m-1} + p_2 q \cdot y^{m-2} + \cdots + p_m q^{m-1} \in \mathbb{Z}[x_1, \ldots, x_k, y]$$

(multiply (4.2) by q^m). We have

$$F = \mathbb{Q}(x_1, \ldots, x_k)[y]/\langle r(y) \rangle$$

where $\langle r(y) \rangle = \{a \cdot r \mid a \in \mathbb{Q}(x_1, \ldots, x_k)[y]\}$ is the ideal generated by r.

Each of the finitely many generators of the group G is a matrix, whose entries are polynomials in the variable y with coefficients from the fraction field $\mathbb{Q}(x_1, \ldots, x_k)$. Every such coefficient is a fraction $a(x_1, \ldots, x_k)/b(x_1, \ldots, x_k)$ with $a(x_1, \ldots, x_k)$, $b(x_1, \ldots, x_k) \in \mathbb{Z}[x_1, \ldots, x_k]$. Let $g(x_1, \ldots, x_k)$ be the least common multiple of all denominators $b(x_1, \ldots, x_k)$, which is a fixed polynomial. Instead of asking whether $A_1 \cdots A_n \equiv \mathsf{Id} \bmod q(y)$ (for group generators $A_1, \ldots, A_n$ of G) we can ask whether $gA_1 \cdots gA_n \equiv g^n \mathsf{Id} \bmod q(y)$.[2] So far, the proof has been following more or less closely Waack's arguments from [89].

Let $M_i = gA_i$, which is a triangular matrix of dimension d for some fixed $d \in \mathbb{N}$ with entries from $\mathbb{Z}[x_1, \ldots, x_k, y]$. Let us write $M_i = D_i + U_i$ where D_i is a diagonal matrix and U_i is triangular with all diagonal entries equal to zero. We get

$$M_1 \cdots M_n = \prod_{i=1}^{n} (D_i + U_i) = \sum_{X_1 \in \{D_1, U_1\}} \cdots \sum_{X_n \in \{D_n, U_n\}} \prod_{j=1}^{n} X_j. \tag{4.3}$$

If there are more than $d-1$ factors U_i in a product $\prod_{j=1}^{n} X_j$, then the product is the zero matrix. So there are at most $\sum_{i=0}^{d-1} \binom{n}{i} \leq d\binom{n}{d} \leq dn^d$ summands (for $n > 2d$) in the sum (4.3) that are not equal to zero. When we look at one of the products $\prod_{j=1}^{n} X_j$ with at most $d-1$ many factors U_i, we can write it as

$$\begin{aligned} &\left(\textstyle\prod_{i=1}^{m_1-1} D_i\right) U_{m_1} \left(\textstyle\prod_{i=m_1+1}^{m_2-1} D_i\right) \cdots U_{m_l} \left(\textstyle\prod_{i=m_l+1}^{n} D_i\right) \\ &\quad = D_{1,m_1-1} U_{m_1} D_{m_1+1,m_2-1} \cdots U_{m_l} D_{m_l+1,n} \end{aligned}$$

[1] Every finitely generated extension field of a perfect field has a separating transcendence base and every prime field is perfect.

[2] Here, for two $(d \times d)$-matrices A and B, $A \equiv B \bmod q(x)$ means that $A[i,j] \equiv B[i,j] \bmod q(x)$ for all $1 \leq i, j \leq d$.

for some $0 \leq l \leq d-1$ and $1 \leq m_1 < \cdots < m_l \leq n$ where $D_{u,v} = \prod_{i=u}^{v} D_i$ ($1 \leq u \leq v+1$, $0 \leq v \leq n$) is a product of at most n diagonal matrices. Each of these products can be calculated by calculating d products of at most n polynomials from $\mathbb{Z}[x_1, \ldots, x_k, y]$, which can be done in DLOGTIME-uniform TC^0 by Lemma 4.3. Moreover, all products $D_{u,v}$ for $1 \leq u < v \leq n$ can be computed in parallel. Once these products are computed, we can, in parallel, compute for all $0 \leq l \leq d-1$ and $1 \leq m_1 < \cdots < m_l \leq n$ the matrix product $D_{1,m_1-1}U_{m_1}D_{m_1+1,m_2-1}\cdots U_{m_l}D_{m_l+1,n}$. Note that these products have constant length and hence involve a constant number of polynomial multiplications and additions. So, all the above matrix products can be computed in DLOGTIME-uniform TC^0 as well. Next, we have to compute the sum of all polynomially many matrices computed in the previous step. For this we have to compute d^2 many sums of polynomially many polynomials, which is again possible in DLOGTIME-uniform TC^0. The resulting matrix is $M_1 \cdots M_n = g^n A_1 \cdots A_n$. Finally we have to reduce all entries of the matrices $M_1 \cdots M_n$ and $g^n\mathsf{Id}$ modulo the minimal polynomial $q(y)$ which can also be done in DLOGTIME-uniform TC^0 by Lemma 4.4. Note that we divide by the polynomial $q(y)$, whose leading coefficient is indeed 1.

We have shown that the word problem for a f.g. solvable linear group G belongs to DLOGTIME-uniform TC^0. If G is in addition infinite, then it cannot be a torsion-group, since every f.g. linear torsion group is finite by a result of Schur, see [90, Corollary 4.9]. Therefore $\mathbb{Z}$ is a subgroup of G. Since the word problem for $\mathbb{Z}$ is already complete for DLOGTIME-uniform TC^0 (it corresponds to the problem of counting the number of ones in a string) we obtain the lower bound in the theorem.

Finally, let G be a f.g. virtually solvable linear group G. Then G contains a normal solvable subgroup H, for which we know that the word problem can be solved in DLOGTIME-uniform TC^0. Moreover, the quotient G/H is a finite group, for which the word problem belongs to DLOGTIME-uniform NC^1. Hence, Theorem 4.5 implies that the word problem for G belongs to DLOGTIME-uniform NC^1.

□

By Tits alternative [82], every linear group is either virtually solvable or contains a free group of rank two. Since by [75, Theorem 6.3], the word problem for a free group of rank two is hard for DLOGTIME-uniform NC^1, one gets the following result:

Theorem 4.6. *For every f.g. linear group that is not virtually solvable, the word problem is hard for* DLOGTIME*-uniform* NC^1*.*

Theorem 4.2 and Theorem 4.6 leave open the case of a f.g. linear group G that is not solvable but a finite extension of a solvable group H. If the quotient G/H is solvable too, then G is solvable and we can apply Theorem 4.2. So, we can assume that the finite quotient G/H is not solvable. It seems plausible that in this case, the word problem for G is hard for DLOGTIME-uniform NC^1, since the word problem for every finite non-solvable group is hard for DLOGTIME-uniform NC^1 [17]. But it is not clear, whether the word problem for the finite quotient G/H reduces to the word problem for G.

Chapter 5

The circuit evaluation problem

5.1 Introduction

There is a natural generalization from the word problem to general algebraic structures: let $\mathcal{A}$ be an algebraic structure with a generating set I. Given two expressions e_1, e_2 built up from the elements of I using the operations f_i, do e_1 and e_2 evaluate to the same element of $\mathcal{A}$. In this chapter we will consider this problem for a more succinct representation of elements: algebraic circuits. An algebraic circuit can be seen as a compressed algebraic expression with a possibly exponential compression ratio. E.g. for a binary function f and an input value a we define the expression e_n via $e_1 = f(a, a)$ and $e_i = f(e_{i-1}, e_{i-1})$ for $2 \leq i \leq n$. Then the expression e_n has length 2^n but can be represented by a circuit if size n. One of the first decision problems for circuits, whose complexity has been studied, is the circuit evaluation problem for Boolean circuits, which was shown to be P-complete by Ladner [52]. In this chapter the circuit evaluation problem will be defined for general algebraic structures, based on the word problem, as the problem to decide for two given circuits whether they evaluate to the same element. For various algebraic structures many results are already known about this problem. First we will formally define the problem and give an overview over various formulations of this problem. Then we take a closer look on the circuit evaluation problem for various structures, namely the ring $(\mathbb{Z}, +, \cdot)$, polynomial rings, finite structures and groups. The results for the ring $(\mathbb{Z}, +, \cdot)$ can mainly be seen as auxiliary results, since for many more complex structures circuit evaluation can be reduced to a circuit over this ring. Then for the other structures known results are considered and brought into context with the results that are shown in this thesis.

5.2 Algebraic circuits

Definition 5.1 (circuit)**.** *A circuit over a f.g. algebraic structure $\mathcal{A} = (D, f_1, \ldots, f_k)$ where the f_i $(1 \leq i \leq k)$ are mappings $f_i : D^{n_i} \to D$ and I is a finite generating set of $\mathcal{A}$, is a triple $\mathcal{C} = (V, S, \mathsf{rhs})$ where*

- *V is a finite set of* gates*,*
- *$S \in V$ is the* output gate *and*
- rhs *(which stands for right-hand side) is a function that assigns to each gate $A \in V$ an element $a \in I$ or an expression of the form $f_i(A_1, \ldots, A_{n_i})$ such that the binary relation $\{(A, B) \in V \times V \mid A \text{ occurs in } \mathsf{rhs}(B)\}$ is acyclic.*

For a gate $A \in V$ with $\mathsf{rhs}(A) = f_i(A_1, \ldots, A_{n_i})$, the gates $A_1, \ldots A_{n_i} \in V$ are called the *input gates for A*. The reflexive and transitive closure of the relation

$$\{(A, B) \in V \times V \mid A \text{ is an input gate for } B\}$$

is a partial order on V that we denote by $\leq_{\mathcal{C}}$.

Definition 5.2 (value of a circuit)**.** *Let $\mathcal{C} = (V, S, \mathsf{rhs})$ be a circuit over $\mathcal{A} = (D, f_1, \ldots, f_k)$ and I be a finite generating set of $\mathcal{A}$. Every gate $A \in V$ evaluates to an element $[A]_{\mathcal{C}} \in D$ in the natural way: if $\mathsf{rhs}(A) = a \in I$, then $[A]_{\mathcal{C}} = a$ and if $\mathsf{rhs}(A) = f_i(A_1, \ldots, A_{n_i})$, then $[A]_{\mathcal{C}} = f_i([A_1]_{\mathcal{C}}, \ldots, [A_{n_i}]_{\mathcal{C}})$. Moreover, we define $[\mathcal{C}] = [S]_{\mathcal{C}}$ (the value computed by $\mathcal{C}$). If the circuit $\mathcal{C}$ is clear from the context, we write $[A]$ instead of $[A]_{\mathcal{C}}$.*

We say that two circuits $\mathcal{C}_1, \mathcal{C}_2$ over the same algebraic structure are *equivalent* if $[\mathcal{C}_1] = [\mathcal{C}_2]$. Sometimes we also use circuits without an output gate; such a circuit is just a pair (V, rhs). A subcircuit of $\mathcal{C}$ is the restriction of $\mathcal{C}$ to a downwards closed (w.r.t. $\leq_{\mathcal{C}}$) subset of V. A gate A with $\mathsf{rhs}(A) = f_i(A_1, \ldots, A_{n_i})$ is called an *inner gate*, otherwise it is an *input gate* of $\mathcal{C}$. Quite often, we view a circuit as a directed acyclic graph (dag) where the inner nodes are labeled with some $f_i (1 \leq i \leq k)$, and the leaf nodes are labeled with elements of I.

The *circuit evaluation problem* for the finitely generated algebraic structure $\mathcal{A}$ ($\mathsf{CEP}(\mathcal{A})$) is the following decision problem:

Input: Two circuits $\mathcal{C}_1 = (V_1, S_1, \mathsf{rhs}_1)$ and $\mathcal{C}_2 = (V_2, S_2, \mathsf{rhs}_2)$ over $\mathcal{A}$.
Question: Does $[\mathcal{C}_1] = [\mathcal{C}_2]$ hold?

The input size is $|V_1| + |V_2|$. Since there is a fixed finite generating set $I \subseteq D$, we can assume that every input value of the circuit has constant size. Note that the complexity of the problem does not depend on the chosen input set I, since for two generating input sets I_1 and I_2 one can transform every element in I_1 into a fixed sized expression with elements from I_2 and vice versa.

For most algebraic structures investigated in the following chapters a slightly different but equivalent version of the circuit evaluation problem is used:

- For a unitary ring $(R, +, \cdot)$ we consider the following problem:

  Input: A circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over $(R, +, \cdot)$.
  Question: Does $[\mathcal{C}] = 0$ hold?

- For a group $(G, \cdot)$ we consider the following problem:

  Input: A circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over $(G, \cdot)$.
  Question: Does $[\mathcal{C}] = 1$ hold?

- For a finite algebraic structure $\mathcal{A}$ we consider the following problem:

  Input: A circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over $\mathcal{A}$ and an element $a \in D$ from its domain.
  Question: Does $[\mathcal{C}] = a$ hold?

Note that the problems above are AC^0-equivalent to the circuit evaluation problem: on the one hand we can consider $0, 1$ resp. $a \in D$ as a circuit $\mathcal{C}_2$. On the other hand assume there are two circuits $\mathcal{C}_1 = (V_1, S_1, \mathsf{rhs}_1)$ and $\mathcal{C}_2 = (V_2, S_2, \mathsf{rhs}_2)$ over $\mathcal{A}$.

- In the case $\mathcal{A}$ is a unitary ring $(R, +, \cdot)$ define a new circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ via $V = V_1 \cup V_2 \cup \{S, S'\}$ and $\mathsf{rhs}(S) = S_1 + S'$, $\mathsf{rhs}(S') = (-1) \cdot S_2$, and $\mathsf{rhs}(A) = \mathsf{rhs}_i(A)$ for $A \in V_i$ $(i \in \{1, 2\})$. Then $[\mathcal{C}_1] = [\mathcal{C}_2]$ if and only if $[\mathcal{C}] = 0$.

- In the case $\mathcal{A}$ is a group define a new circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ via $V = V_1 \cup V_2 \cup \{S\}$ and $\mathsf{rhs}(S) = S_1 \cdot S_2$ and

$$\mathsf{rhs}(A) = \begin{cases} \mathsf{rhs}_1(A) \text{ if } A \in V_1 \\ C \cdot B \text{ if } A \in V_2 \text{ and } \mathsf{rhs}_2(A) = B \cdot C \\ a^{-1} \text{ if } A \in V_2 \text{ and } \mathsf{rhs}_2(A) = a \in I \text{ (where } I \text{ is a finite generating set)} \end{cases} .$$

  Then $[S_2]_{\mathcal{C}} = ([S_2]_{\mathcal{C}_2})^{-1}$ and $[\mathcal{C}_1] = [\mathcal{C}_2]$ if and only if $[\mathcal{C}] = 1$.

- In the case $\mathcal{A}$ is a finite structure we can test in parallel for all elements $a \in D$ whether $[\mathcal{C}_1] = a$ and $[\mathcal{C}_2] = a$. This is true for an $a \in D$ if and only if $[\mathcal{C}_1] = [\mathcal{C}_2]$.

Clearly, for every finite structure the circuit evaluation problem can be solved in polynomial time by evaluating all gates along the partial order $\leq_{\mathcal{C}}$. For the sake of convenience we sometimes call the circuit evaluation problem for $\mathcal{A}$ simply "circuit evaluation for $\mathcal{A}$".

From now on we consider mostly circuits over a semiring $(R, +, \cdot)$ or a semigroup $(S, \cdot)$ with a generating set I. In our proofs it is sometimes convenient to allow arbitrary terms built from $V \cup I$ using $+$ and $\cdot$ in right-hand sides and gates of the form $\mathsf{rhs}(A) = B$ where B is a gate again (so-called copy gates). For instance, we might have $\mathsf{rhs}(A) = s \cdot B \cdot t + C + s$ for $s, t \in I$ and $B, C \in V$. We also say that circuits where all right-hand sides are of the form $A + B$, $A \cdot B$, or $a \in I$, are in *normal form*. We will make use of the following fact:

Lemma 5.3. *A given circuit over a semiring $(R, +, \cdot)$ can be transformed in logarithmic space into an equivalent normal form circuit.*

Proof. The only non-trivial part is the elimination of copy gates A with $\mathsf{rhs}(A) = B$ for a gate B; all other right-hand sides that violate the normal form have to be split up using fresh gates. This is easily done in logarithmic space. For copy gates consider the directed graph G that contains for every copy gate A the gate A as well as the gate $\mathsf{rhs}(A)$. Moreover, there is a directed edge from A to $B = \mathsf{rhs}(A)$. This is a directed forest where the edges are oriented towards the roots since every node has at most one outgoing edge (and the graph is acyclic). By traversing all (deterministic) paths, we can compute the reflexive transitive closure G^* of G in logarithmic space. Using G^* it is straightforward to eliminate copy gates: for every copy gate A we redefine $\mathsf{rhs}(A) = \mathsf{rhs}(B)$ where B is the unique node in G^* of outdegree zero such that (A, B) is an edge of G^*. □

A gate A where $\mathsf{rhs}(A)$ has the form $B + C$ (resp., $B \cdot C$) is called an addition gate (resp., multiplication gate). The *depth* $\mathsf{depth}(A)$ (resp., *multiplication depth* $\mathsf{mdepth}(A)$) of the gate A is the maximal number of gates (resp., multiplication gates) along a path from an input gate to A. So, input gates have depth one and multiplication depth zero. The *depth* (resp., *multiplication depth*) of $\mathcal{C}$ is $\mathsf{depth}(\mathcal{C}) = \mathsf{depth}(S)$ (resp., $\mathsf{mdepth}(\mathcal{C}) = \mathsf{mdepth}(S)$). The *formal degree* $\mathsf{deg}(A)$ of a gate A is 1 if A is an input gate, $\max\{\mathsf{deg}(B), \mathsf{deg}(C)\}$ if $\mathsf{rhs}(A) = B + C$, and $\mathsf{deg}(B) + \mathsf{deg}(C)$ if $\mathsf{rhs}(A) = B \cdot C$. The formal degree of $\mathcal{C}$ is $\mathsf{deg}(\mathcal{C}) = \mathsf{deg}(S)$.

5.3 Circuit evaluation for $(\mathbb{Z}, +, \cdot)$: some auxiliary results

It is known that in general circuit evaluation for the ring $(\mathbb{Z}, +, \cdot)$ is in coRP but it is still not known if there exists a polynomial time algorithm for this problem, see e.g. [6]. We will consider this problem in detail later in Section 5.6 . In this section we mainly investigate circuits with bounded multiplication depth resp., bounded formal degree. The results of this section have appeared in [1]. The main results are the following:

- If the multiplication depth is bounded by a constant, then the multiplication gates can be eliminated and the circuit can be evaluated in $\mathsf{C}_=\mathsf{L}$.
- If the formal degree is polynomially bounded, then circuit evaluation is in $\mathsf{C}_=\mathsf{LogCFL}$.

From now on for circuits over $(\mathbb{Z}, +, \cdot)$ we fix $\{-1, 0, 1\}$ as the input set.

Definition 5.4 (positive circuit, addition circuit)**.**

- *A positive circuit is a circuit over $(\mathbb{Z}, +, \cdot)$ without input gates labeled by the constant -1.*
- *An addition circuit is a positive circuit without multiplication gates.*

Lemma 5.5. *Given a circuit $\mathcal{C}$ over $(\mathbb{Z}, +, \cdot)$ one can compute in logarithmic space two positive circuits $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $[\mathcal{C}] = [\mathcal{C}_1] - [\mathcal{C}_2]$. Moreover, for $i \in \{1, 2\}$ we have $\mathsf{deg}(\mathcal{C}_i) \leq \mathsf{deg}(\mathcal{C})$, $\mathsf{depth}(\mathcal{C}_i) \leq 2 \cdot \mathsf{depth}(\mathcal{C})$, and $\mathsf{mdepth}(\mathcal{C}_i) \leq \mathsf{mdepth}(\mathcal{C})$.*

Proof. Let $\mathcal{C} = (V, S, \mathsf{rhs})$ be a circuit over $(\mathbb{Z}, +, \cdot)$. We define the positive circuits $\mathcal{C}_1 = (V', S_1, \mathsf{rhs}')$ and $\mathcal{C}_2 = (V', S_2, \mathsf{rhs}')$ as follows:

- $V' = \{A_i \mid A \in V, i \in \{1, 2\}\}$,
- $\mathsf{rhs}'(A_i) = B_i + C_i$ if $\mathsf{rhs}(A) = B + C$ for $i \in \{1, 2\}$,
- $\mathsf{rhs}'(A_1) = B_1C_1 + B_2C_2$ if $\mathsf{rhs}(A) = B \cdot C$,
- $\mathsf{rhs}'(A_2) = B_1C_2 + B_2C_1$ if $\mathsf{rhs}(A) = B \cdot C$,
- $\mathsf{rhs}'(A_1) = \mathsf{rhs}(A)$ if $\mathsf{rhs}(A) \in \{0, 1\}$,
- $\mathsf{rhs}'(A_2) = 0$ if $\mathsf{rhs}(A) \in \{0, 1\}$,
- $\mathsf{rhs}'(A_1) = 0$ if $\mathsf{rhs}(A) = -1$,
- $\mathsf{rhs}'(A_2) = 1$ if $\mathsf{rhs}(A) = -1$.

Now we show by induction that for every gate $A \in V$ we have $[A] = [A_1] - [A_2]$: the case that A is an input gate is trivial. Now let A be an addition gate with $\mathsf{rhs}(A) = B + C$ and the statement be true for B and C. Then

$$\begin{aligned}[A] &= [B] + [C] \\ &= [B_1] - [B_2] + [C_1] - [C_2] \\ &= ([B_1] + [C_1]) - ([B_2] + [C_2]) \\ &= [A_1] - [A_2]\end{aligned}$$

Finally, let A be a multiplication gate with $\mathsf{rhs}(A) = B \cdot C$ and the statement be true for B and C. Then

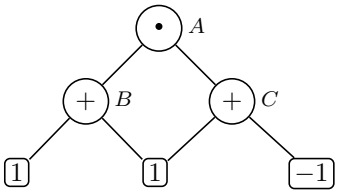
$$\begin{aligned}[A] &= [B][C] \\ &= ([B_1] - [B_2])([C_1] - [C_2]) \\ &= [B_1][C_1] + [B_2][C_2] - [B_1][C_2] - [B_2][C_1] \\ &= [A_1] - [A_2].\end{aligned}$$

So the claim holds. The construction of $\mathcal{C}_1$ and $\mathcal{C}_2$ can be done in logarithmic space. By induction, it can be shown that for every gate $A \in V$ and every $i \in \{1, 2\}$, one has $\mathsf{deg}(A_i) = \mathsf{deg}(A)$, $\mathsf{depth}(A_i) \leq 2 \cdot \mathsf{depth}(A)$, and $\mathsf{mdepth}(A_i) = \mathsf{mdepth}(A)$. □

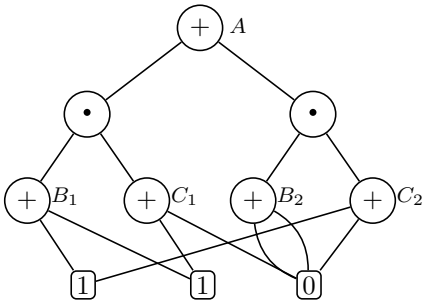
Example 5.6. *Figure 5.1 illustrates the proof of Lemma 5.5. The circuit $\mathcal{C}$ evaluates to $(1 + 1) \cdot (1 - 1) = 1 + 1 - 1 - 1 = 2 - 2 = 0$. The two circuits $\mathcal{C}_1$ and $\mathcal{C}_2$ that are constructed as described in the proof evaluate to the positive part ($[\mathcal{C}_1] = 2 + 0 = 2$) and negative part ($[\mathcal{C}_2] = 0 + 2 = 2$) of this sum.*

Definition 5.7 (structure-preserving partition)**.** *A partition $\biguplus_{i=1}^{m} V_i$ of the set of all multiplication gates of $\mathcal{C}$ is called structure-preserving if for all multiplication gates u, v of $\mathcal{C}$ the following holds: if there is a non-empty path from u to v in (the dag corresponding to) $\mathcal{C}$ then there exist $1 \leq i < j \leq d$ such that $u \in V_i$ and $v \in V_j$.*

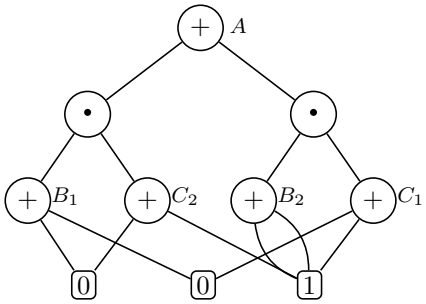
Lemma 5.8. *Let d be constant. From a given positive circuit $\mathcal{C}$ of multiplication depth d together with a structure-preserving partition $\biguplus_{i=1}^{d} V_i$, we can compute in logarithmic space an addition circuit $\mathcal{D}$ such that $[\mathcal{C}] = [\mathcal{D}]$.*



(a) Circuit $\mathcal{C}$ over $(\mathbb{Z}, +, \cdot)$.



(b) Circuit $\mathcal{C}_1$



(c) Circuit $\mathcal{C}_2$

Figure 5.1: Transformation of a circuit $[\mathcal{C}]$ into two positive circuits $[\mathcal{C}_1]$ and $[\mathcal{C}_2]$ such that $[\mathcal{C}_1] - [\mathcal{C}_2] = [\mathcal{C}]$.

Proof. Let $\mathcal{C} = (V, S, \mathsf{rhs})$ together with a structure-preserving partition $\biguplus_{i=1}^{d} V_i$. W.l.o.g. we can assume that there are two unique input gates whose right-hand sides are 0, resp. 1 and we denote these gates simply by 0, resp., 1.

Since d is a constant, it suffices to construct in logarithmic space a positive circuit $\mathcal{C}' = (V', S, \mathsf{rhs}')$ of multiplication depth $d-1$ together with a structure-preserving partition $\biguplus_{i=1}^{d-1} V_i'$ of the set of all multiplication gates of $\mathcal{C}'$ such that $[\mathcal{C}] = [\mathcal{C}']$ (the composition of a constant number of logspace computations is again a logspace computation).

To achieve the above goal, we eliminate in $\mathcal{C}$ all multiplication gates from V_1. Note that below these gates there are no other multiplication gates. Then, we define the set V_i' as V_{i+1} for $1 \leq i \leq d-1$.

Let $V_1 = \{A_1, \ldots, A_m\}$ and assume that $\mathsf{rhs}_{\mathcal{C}}(A_i) = B_i \cdot C_i$. The set of gates of $\mathcal{C}'$ is

$$V' = V \cup \{A^{(i)} \mid A \in V, 1 \leq i \leq m\},$$

i.e., we add m copies of each gate to the circuit. We define the right-hand side mapping as follows:

$$\mathsf{rhs}'(A) = \mathsf{rhs}(A) \text{ if } A \in V \setminus V_1 \tag{5.1}$$

$$\mathsf{rhs}'(A_i) = B_i^{(i)} \text{ for } 1 \leq i \leq m \tag{5.2}$$

$$\mathsf{rhs}'(A^{(i)}) = B^{(i)} + C^{(i)} \text{ if } A \in V \text{ and } \mathsf{rhs}(A) = B + C \tag{5.3}$$

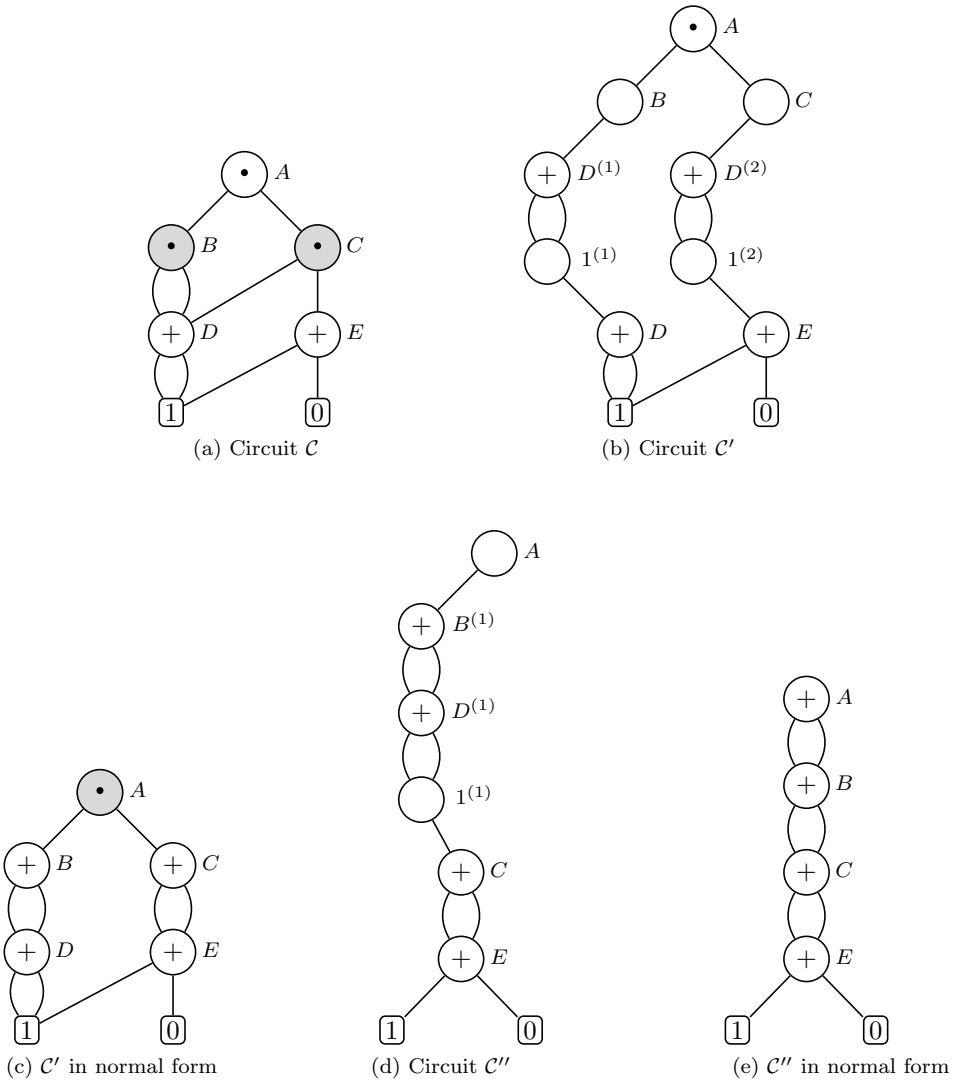
$$\mathsf{rhs}'(A^{(i)}) = 0 \text{ if } A \in V \text{ and } \mathsf{rhs}(A) = B \cdot C \tag{5.4}$$

$$\mathsf{rhs}'(0^{(i)}) = 0 \tag{5.5}$$

$$\mathsf{rhs}'(1^{(i)}) = C_i \tag{5.6}$$

The idea of the above construction is the following: basically, we add m copies of the circuit $\mathcal{C}$. In these copies, we do not need the multiplication gates[1] and since we do not want to introduce new multiplication gates, we set the right-hand side of a copy of a multiplication gate to 0, see

[1] Actually, we only need in the i-th copy those nodes that belong to a path from the unique 1-gate to B_i. But we cannot compute the set of these nodes in logarithmic space unless $\mathsf{L} = \mathsf{NL}$. Hence, we put all nodes into the copy.



(a) Circuit $\mathcal{C}$ (b) Circuit $\mathcal{C}'$

(c) $\mathcal{C}'$ in normal form (d) Circuit $\mathcal{C}''$ (e) $\mathcal{C}''$ in normal form

Figure 5.2: Illustration of the proof of Lemma 5.8 with the structure-preserving partition $V_1 = \{B, C\}$ and $V_2 = \{A\}$.

(5.4).[2] Also notice that strictly below A_i we only find addition gates and constants in the circuit $\mathcal{C}$. In particular, the value $[B_i]_{\mathcal{C}}$ is equal to the number of paths from the unique 1-gate 1 to B_i and similarly for C_i. We want to assign to the gate A_i the product of these path numbers. For this, we redirect the edges (B_i, A_i) and (C_i, A_i) of the multiplication gate for every $1 \leq i \leq m$ as follows: the edge (C_i, A_i) is replaced by the edge $(C_i, 1^{(i)})$, see (5.6). Moreover, the edge (B_i, A_i) is replaced by the edge $(B_i^{(i)}, A_i)$ (which is the unique incoming edge to A_i), see (5.2). So, basically, we serially connect the circuit part between 1 and C_i with the circuit part between 1 and B_i. Thereby we multiply the number of paths. The above construction can be clearly done in logarithmic space.

□

Example 5.9. *Figure 5.2 is an example for the construction of an addition circuit from a positive circuit with multiplication depth 2. On picture (a) the circuit is shown and the set* $V_1 = \{B, C\}$

[2] This is an arbitrary choice; instead of 0 we could have also taken 1.

is marked grey. In the first step we construct $\mathcal{C}'$ as described in the proof. Copy-gates are left unlabeled. On picture (b) is shown the part of $\mathcal{C}'$ that is connected to the inputs. Recall that we do not know this part during the construction, but for reasons of clarity we do not show the other part in the illustration. In picture (c) $\mathcal{C}'$ is transformed into normal form, the gates are renamed and $V_2 = \{A\}$ is marked grey. Then we repeat the previous two steps to reach the final addition circuit that is shown in picture (e).

Recall that the following problem is $\mathsf{C}_=\mathsf{L}$-complete: the input consists of two dags G_1 and G_2 and vertices s_1, t_1 (in G_1) and s_2, t_2 (in G_2), and it is asked whether the number of different paths from s_1 to t_1 in G_1 is equal to the number of different paths from s_2 to t_2 in G_2. This problem is easily seen [84] to be equivalent to the following problem: given two addition circuits $\mathcal{C}_1$ and $\mathcal{C}_2$, does $[\mathcal{C}_1] = [\mathcal{C}_2]$ hold? Combining this with Lemma 5.5 and Lemma 5.8 leads to the following result:

Lemma 5.10. *Let d be a constant. For a given circuit $\mathcal{C}$ over $(\mathbb{Z}, +, \cdot)$ of multiplication depth d together with a structure-preserving partition $\biguplus_{i=1}^{d} V_i$ of the set of all multiplication gates of $\mathcal{C}$, the question whether $[\mathcal{C}] = 0$ is $\mathsf{C}_=\mathsf{L}$-complete.*

With the same argumentation as for circuits over $(\mathbb{Z}, +)$ one gets the following result:

Lemma 5.11. *Given a circuit $\mathcal{C}$ over $(\mathbb{N}, +)$ one can compute the value of $\mathcal{C}$ in $\#\mathsf{L} \subseteq \mathsf{DET}$.*

In [86] it is shown that $\#\mathsf{LogCFL}$ is the class of all functions $f : \{0,1\}^* \to \mathbb{N}$ (a non-binary input alphabet Σ has to be encoded into $\{0,1\}^*$) for which there exists a logspace-uniform family $(\mathcal{C}_n)_{n\geq 1}$ of circuits over $(\mathbb{N}[x_1, \ldots, x_n], +, \cdot)$ such that $\mathcal{C}_n$ computes the mapping f restricted to $\{0,1\}^n$ and there is a polynomial $p(n)$ such that the formal degree of $\mathcal{C}_n$ is bounded by $p(n)$. So, in particular, for two given such circuit families $(\mathcal{C}_{1,n})_{n\geq 1}$, $(\mathcal{C}_{2,n})_{n\geq 1}$ the question whether $[\mathcal{C}_{1,n}] = [\mathcal{C}_{2,n}]$ for every $n \geq 1$ is $\mathsf{C}_=\mathsf{LogCFL}$-complete. We need the following lemma:

Lemma 5.12. *There is an NAuxPDA $\mathcal{P}$ that gets as input a positive circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over $(\mathbb{Z}, +, \cdot)$ and such that the number of accepting computations of $\mathcal{P}$ on input $\mathcal{C}$ is $[\mathcal{C}]$. Moreover, the running time is bounded polynomially in $\mathsf{depth}(\mathcal{C}) \cdot \mathsf{deg}(\mathcal{C})$.*

Proof. The NAuxPDA $\mathcal{P}$ stores a sequence of gates on its pushdown (every gate can be encoded using $\log(|V|)$ many bits). In the first step it pushes the output gate S on the initially empty pushdown. If A is on top of the pushdown and $\mathsf{rhs}(A) = B+C$, then $\mathcal{P}$ replaces A on the pushdown by B or C, where the choice is made nondeterministically. If $\mathsf{rhs}(A) = B \cdot C$, then $\mathcal{P}$ replaces A on the pushdown by BC. If $\mathsf{rhs}(A) = 0$, then $\mathcal{P}$ terminates and rejects. Finally, if $\mathsf{rhs}(A) = 1$, then $\mathcal{P}$ pops A from the pushdown. If thereby the pushdown becomes empty then $\mathcal{P}$ terminates and accepts. In addition to its pushdown, $\mathcal{P}$ only needs a logarithmic space bounded work tape to store a single gate. Moreover, if we start $\mathcal{P}$ with only the gate A on the pushdown, then (i) the number of accepting computation paths from that configuration is exactly $[A]_{\mathcal{C}}$ (ii) the number of pushdown operations along a computation path is bounded by $\mathsf{depth}(A) \cdot \mathsf{deg}(A)$. Both statements follow by induction. □

Together with Lemma 5.5 this leads immediately to the following result:

Lemma 5.13. *Let p be a fixed polynomial. For a given circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over $(\mathbb{Z}, +, \cdot)$ with formal degree bounded by $p(|V|)$ the question whether $[\mathcal{C}] = 0$ is in $\mathsf{C}_=\mathsf{LogCFL}$.*

5.4 Circuit evaluation for finite structures

As mentioned above for every finite algebraic structure the circuit evaluation problem can be solved in polynomial time. One of the most famous results about circuit evaluation is Ladner's classical P-completeness result for the Boolean circuit value problem [52] that can be stated as follows:

Theorem 5.14 ([52]). *For the Boolean semiring $\mathbb{B}_2 = (\{0,1\}, \vee, \wedge)$, the problem $\mathsf{CEP}(\mathbb{B}_2)$ is P-complete.*

For semigroups, the following fundamental dichotomy was shown in [19]:

Theorem 5.15 ([19]). *Let S be a finite semigroup.*

- *If S is aperiodic, then $\mathsf{CEP}(S)$ is in NL.*
- *If S is solvable, then $\mathsf{CEP}(S)$ belongs to DET.*
- *If S is not solvable, then $\mathsf{CEP}(S)$ is P-complete.*

Some remarks should be made:

- In [19], Theorem 5.15 is only shown for monoids, but the extension to semigroups is straightforward: if the finite semigroup S has a non-solvable subgroup, then $\mathsf{CEP}(S)$ is P-complete, since the circuit evaluation problem for a non-solvable finite group is P-complete. On the other hand, if S is solvable (resp., aperiodic), then also the monoid S^1 is solvable (resp., aperiodic). This holds, since the subgroups of S^1 are exactly the subgroups of S together with $\{1\}$. Hence, $\mathsf{CEP}(S^1)$ is in DET (resp., NL), which implies that $\mathsf{CEP}(S)$ is in DET (resp., NL).
- In [19], the authors use the original definition $\mathsf{DET} = \mathsf{NC}^1(\mathsf{det})$ of Cook. But the arguments in [19] actually show that for a finite solvable semigroup, $\mathsf{CEP}(S)$ belongs to $\mathsf{AC}^0(\mathsf{det})$ (which is our definition of DET).
- In [19], the authors study two versions of the circuit evaluation problem for a semigroup S: what we call $\mathsf{CEP}(S)$ is called $\mathsf{UCEP}(S)$ (for "unrestricted circuit evaluation problem") in [19]. The problem $\mathsf{CEP}(S)$ is defined in [19] as the circuit evaluation problem where in addition the input circuit must have the property that the output gate has no outgoing edges and all gates are connected to the output gate. These conditions can be enforced with an $\mathsf{AC}^0(\mathsf{NL})$-precomputation. Hence, the difference between the two variants is only relevant for classes below NL. We only consider the unrestricted version of the circuit evaluation problem (where the input circuit is arbitrary).

In Chapter 8 we extend this dichotomy to finite semirings and show that if a finite semiring R is $\{0,1\}$-free and $\mathcal{R}^\bullet$ is solvable, then $\mathsf{CEP}(R)$ is in DET. Otherwise it is P-complete.

5.5 Skew circuits over semirings

Definition 5.16 (skew circuit). *A circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over a semiring $(R, +, \cdot)$ is a skew circuit, if for every multiplication gate A at least one of its inputs is an input gate of the circuit.*

For skew circuits $\mathcal{C} = (V, S, \mathsf{rhs})$ over R with a generating set I we will use a quite simplified notation: we assume that the right-hand side of a multiplication gate A is of the form $b \cdot C$ (resp., $C \cdot b$) with $b \in I$ and $C \in V$. Notice that this differs slightly from the definition, since formally $\mathsf{rhs}(A) = B \cdot C$ (resp., $\mathsf{rhs}(A) = C \cdot B$) with $\mathsf{rhs}(B) \in I$ and $C \in V$.

Definition 5.17 (branching program). *A branching program over the semiring $(R, +, \cdot)$ with a finite generating set I is a tuple $\mathcal{B} = (V, E, \lambda, s, t)$ where (V, E) is a directed acyclic graph, $\lambda : E \to I$ assigns to each edge a generating element, and $s, t \in V$. Let $\mathcal{P}$ be the set of all paths from s to t. For a path $p = (v_0, v_1, \ldots, v_n) \in \mathcal{P}$ ($v_0 = s$, $v_n = t$) we define $\lambda(p) = \prod_{i=1}^{n} \lambda(v_{i-1}, v_i)$ as the product of all edge labels along the path. Finally, the value defined by $\mathcal{B}$ is*

$$[\mathcal{B}] = \sum_{p \in \mathcal{P}} \lambda(p).$$

It is well known that for commutative unitary semirings skew circuits and branching programs are basically the same objects. It is also well known that the value defined by a branching program $\mathcal{B}$ over a unitary semiring can be computed using matrix powers over the semiring R^0 (the 0 is needed to fill the matrix with an absorbing identity element in the case that R does not contain an absorbing 0). W.l.o.g. assume that $\mathcal{B} = (\{1, \ldots, n\}, E, \lambda, 1, n)$ and consider the adjacency matrix M of the edge-labeled graph $(\{1, \ldots, n\}, E, \lambda)$, i.e., the $(n \times n)$-matrix M with $M[i, j] = \lambda(i, j)$ for $(i, j) \in E$ and all other entries are equal to 0. Then

$$[\mathcal{B}] = \left(\sum_{i=0}^{n} M^i \right) [1, n].$$

For many commutative unitary semirings R, this simple fact can be used to get an NC^2-algorithm for computing $[\mathcal{B}]$. The $n + 1$ matrix powers M^i $(0 \leq i \leq n)$ can be computed in parallel, and every power can be computed by a balanced tree of height $\log i \leq \log n$, where every tree node computes a matrix product. Hence, we obtain an NC^2-algorithm, if

(i) the number of bits needed to represent a matrix entry in M^n is polynomially bounded in n and in the size of the entries of M.

(ii) the product of two matrices over the semiring R can be computed in NC^1.

Point (ii) holds if products of two elements and iterated sums in R can be computed in NC^1. Notice that the adding of a new additive identity does not change these properties. For instance these facts are well known for the semirings $(\mathbb{Z} \cup \{\infty\}, \min, +)$ and $(\mathbb{Z} \cup \{-\infty\}, \max, +)$ where we assume that integers are given in binary representation.

We have defined circuits in such a way that input gates are labeled by generators from a finite set I, but later we will need to evaluate skew circuits over $(\mathbb{Z} \cup \{-\infty\}, \max, +)$ where the input gates are labeled with binary encoded integers. Notice that this does not violate the finiteness property essentially, since by iterated doubling one can construct an equivalent (standard) circuit over $(\mathbb{Z} \cup \{-\infty\}, \max, +)$ with inputs from $\{-1, 1, -\infty\}$. With the argumentation above these circuits can be evaluated in NC^2.

Summing up this discussion leads to the following lemma:

Lemma 5.18. *The circuit evaluation problem for skew circuits (or branching programs) over the semiring* $(\mathbb{Z} \cup \{\infty\}, \min, +)$ *resp.* $(\mathbb{Z} \cup \{-\infty\}, \max, +)$ *where the inputs are binary encoded integers can be solved in* NC^2.

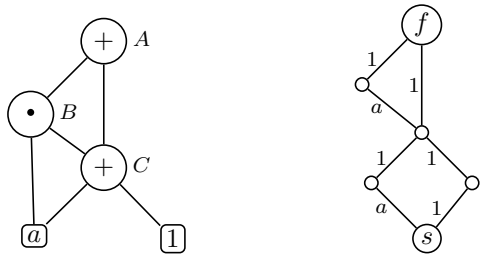
For skew circuits over the polynomial ring $(\mathbb{Z}[x_1, \ldots, x_k], +, \cdot)$ (resp. $(\mathbb{Z}_n[x_1, \ldots, x_k], +, \cdot)$ for some $n \in \mathbb{N}$) for some fixed $k \in \mathbb{N}$ we can even do better and show the following result:

Lemma 5.19. *The circuit evaluation problem for a skew circuit (or branching program) over the polynomial ring* $(\mathbb{Z}[x_1, \ldots, x_k], +, \cdot)$ *(resp.* $(\mathbb{Z}_n[x_1, \ldots, x_k], +, \cdot)$ *for some* $n \in \mathbb{N}$*) for some fixed* $k \in \mathbb{N}$ *can be solved in* $\mathsf{C}_=\mathsf{L}$ *(resp.* DET*).*

Proof. Let $\mathcal{C} = (V, S, \mathsf{rhs})$ be a skew circuit over the polynomial ring $(\mathbb{Z}[x_1, \ldots, x_k], +, \cdot)$ (resp. $(\mathbb{Z}_n[x_1, \ldots, x_k], +, \cdot)$ for some $n \in \mathbb{N}$) for some fixed $k \in \mathbb{N}$ with inputs from $\{-1, 1, x_1, \ldots x_k\}$. W.l.o.g assume that every multiplication gate M is of the form $\mathsf{rhs}(M) = I \cdot A$ with $I \in \{-1, 1, x_1, \ldots, x_k\}$. Furthermore, we can assume that every input gate is either an input of addition gates or an input of multiplication gates (otherwise we copy the gate). With the same argumentation as in the proof of Proposition 2.1 in [6] we know that the polynomial $[\mathcal{C}] = p(x_1, \ldots, x_k)$ is the zero polynomial if and only if $p(\alpha_1, \ldots, \alpha_k) = 0$ for $\alpha_i = 2^{n^{2i}k^i}$. The idea is to replace every occurrence of some x_i by α_i: there are two situations where a variable x_i can appear in $\mathcal{C}$:

1. If x_i is an input of a multiplication gate M, i.e., $\mathsf{rhs}(M) = x_i \cdot A$ for some $A \in V$ we use $n^{2i}k^i$ gates and iterated doubling to add $2^{n^{2i}k^i}$ times the gate A to itself.

2. If x_i is an input of an addition gate, then by iterated doubling we can use $n^{2i}k^i$ gates to add $2^{n^{2i}k^i}$ times the input gate 1 to itself.



(a) Skew Circuit $\mathcal{C}$ over R

(b) Branching program $\mathcal{B}$

Figure 5.3: A skew circuit over some unitary semiring R and the corresponding branching program.

Furthermore we eliminate multiplication gates with input -1 be introducing a gate A' for every gate $A \in V$ such that $[A'] = -[A]$.

Formally we define the purely additive circuit $\mathcal{D} = (V_\mathcal{D}, S, \mathsf{rhs}_\mathcal{D})$ as follows: let $b = n^{2k}k^k$. We set

$$V_\mathcal{D} := V \cup \{A' \mid A \in V\} \cup \{A_i \mid A \in V, 1 \leq i \leq b\} \cup \{A'_i \mid A \in V, 1 \leq i \leq b\} \cup \{X_i \mid 1 \leq i \leq b\}.$$

- $\mathsf{rhs}_\mathcal{D}(A) = B + C$ and $\mathsf{rhs}_\mathcal{D}(A') = B' + C'$ if $\mathsf{rhs}(A) = B + C$
- $\mathsf{rhs}_\mathcal{D}(A) = B$ and $\mathsf{rhs}_\mathcal{D}(A') = B'$ if $\mathsf{rhs}(A) = 1 \cdot B$
- $\mathsf{rhs}_\mathcal{D}(A) = B'$ and $\mathsf{rhs}_\mathcal{D}(A') = B$ if $\mathsf{rhs}(A) = -1 \cdot B$
- $\mathsf{rhs}_\mathcal{D}(A) = B_{n^{2i}k^i}$ if $\mathsf{rhs}(A) = x_i \cdot B$
- $\mathsf{rhs}_\mathcal{D}(A_j) = A_{j-1} + A_{j-1}$ and $\mathsf{rhs}_\mathcal{D}(A'_j) = A'_{j-1} + A'_{j-1}$ for $2 \leq j \leq b$
- $\mathsf{rhs}_\mathcal{D}(A_1) = A + A$ and $\mathsf{rhs}_\mathcal{D}(A'_1) = A' + A'$
- $\mathsf{rhs}_\mathcal{D}(A) = X_{n^{2i}k^i}$ if $\mathsf{rhs}(A) = x_i$
- $\mathsf{rhs}_\mathcal{D}(X_j) = X_{j-1} + X_{j-1}$ for $2 \leq j \leq b$
- $\mathsf{rhs}_\mathcal{D}(X_1) = 1 + 1$
- $\mathsf{rhs}_\mathcal{D}(A) = 1$ and $\mathsf{rhs}_\mathcal{D}(A') = -1$ if $\mathsf{rhs}(A) = 1$
- $\mathsf{rhs}_\mathcal{D}(A) = -1$ and $\mathsf{rhs}_\mathcal{D}(A') = 1$ if $\mathsf{rhs}(A) = -1$

The resulting circuit is purely additive, can be constructed in logarithmic space and it can be shown by induction that if $\mathsf{rhs}(A) = x_i$, then $[A]_\mathcal{D} = \alpha_i$ and if $\mathsf{rhs}(A) = x_i \cdot B$, then $[A]_\mathcal{D} = \alpha_i \cdot [B]$. Overall, if $[\mathcal{C}] = p(x_1, \cdots, x_k)$, then $[\mathcal{D}] = p(\alpha_1, \ldots, \alpha_k)$ and as remarked above $[\mathcal{C}] = 0$ if and only if $[\mathcal{D}] = 0$. By Lemma 5.10 circuit evaluation for $\mathcal{D}$ is in $\mathsf{C}_=\mathsf{L}$. In case $\mathcal{C}$ is a circuit over $\mathbb{Z}_n[x]$ we replace in $\mathcal{D}$ every input -1 by $n-1$ and get a circuit over $\mathbb{Z}_n$ which can be evaluated in DET by Theorem 5.15. □

Example 5.20. *First consider the skew circuit $\mathcal{C}$ over some unitary commutative semiring R with some $a \in R$ in Figure 5.3 to illustrate the algorithm that leads to Lemme 5.18 . At the first step this circuit is transformed into the branching program $\mathcal{B}$ in the same figure. This can be done in the following way:*

1. *First take a new node s and connect s to every input gate with edges that are labeled with the value of the input gate.*

2. *For every gate A with $\mathsf{rhs}(A) = B + C$, label the edges from A to B and from A to C with 1.*

3. *For every gate with $\mathsf{rhs}(A) = i \cdot B$ where i is an input gate delete the edge from A to i and label the edge from A to B with the value of i.*

4. *The output gate of the circuit becomes the node f of the branching program.*

After these steps we get $[\mathcal{C}] = [\mathcal{B}]$. The adjacency matrix of $\mathcal{B}$ is

$$M = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & a \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Since M^5 and M^6 are the zero matrix we obtain

$$[\mathcal{C}] = [\mathcal{B}] = (M^0 + M^1 + M^2 + M^3 + M^4)[1, 6] = 0 + 0 + 0 + (a + 1) + (a^2 + a) = a^2 + 2a + 1.$$

To illustrate the proof of Lemma 5.19 we replace the element a in the circuit above by x and consider $\mathcal{C}$ as a circuit over $\mathbb{Z}[x]$. This circuit is shown in Figure 5.4 in picture (a). Since $|V| = 5$ we know that for $\alpha = 2^{5^2} = 2^{25}$ we get $[\mathcal{C}] = p(x) = 0$ if and only if $p(\alpha) = 0$. In a first step in picture (b) we replace the input x of the $+$-gate C by the gate X_{25} that evaluates to 2^{25} by 25 doubling steps (starting with 1). In the end in picture (c) we replace the input x of the multiplication gate B ($[B] = x \cdot [C]$) by the gate C_{25} that evaluates to $\alpha \cdot [C]$ by 25 doubling steps (starting with C). The resulting circuit is purely additive and evaluates to $p(\alpha)$.

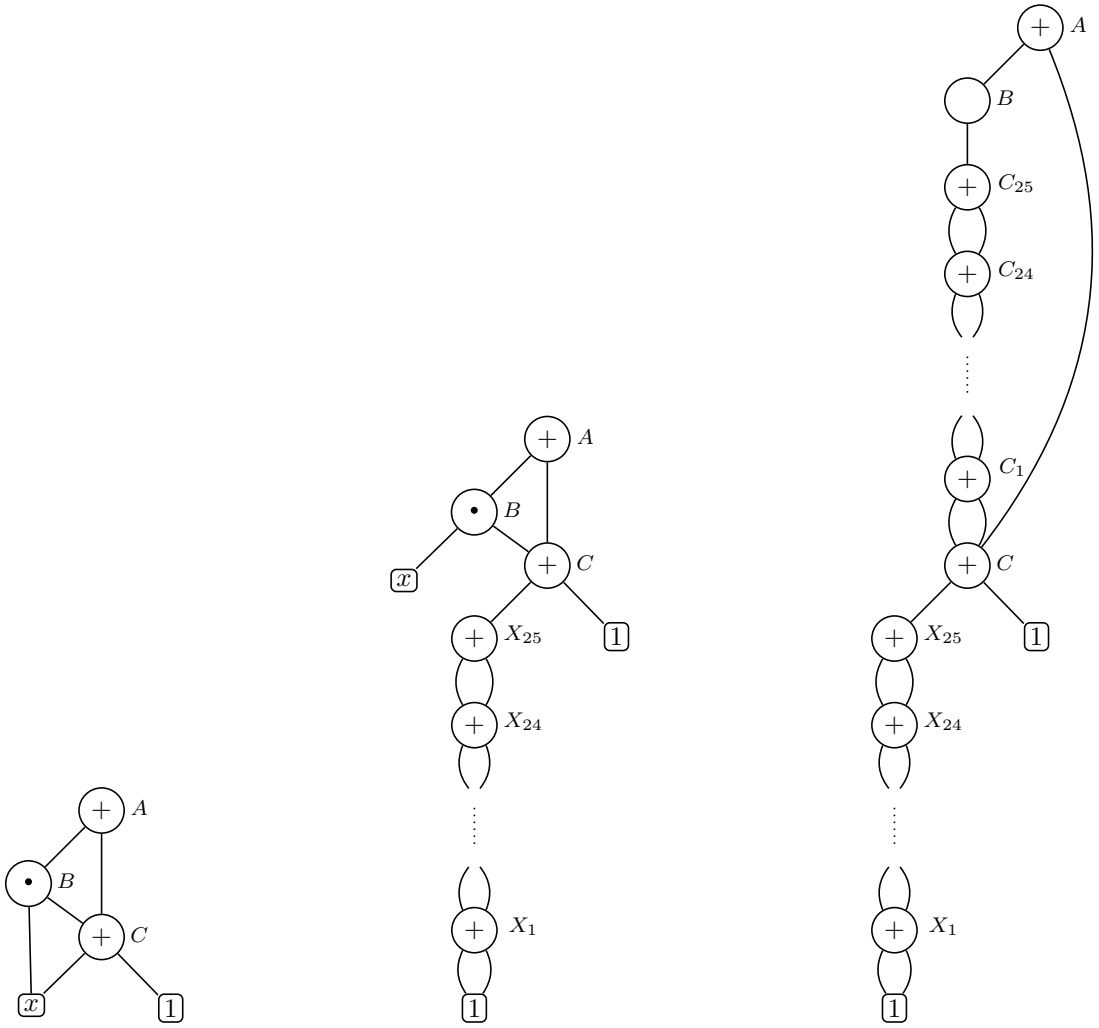
Finally, we show that in skew circuits over $\mathbb{Z}[x]$ we can assume that the inputs are given as polynomials of the form $\sum_{i=1}^{k} a_i x^i$ where the a_i are given in binary encoding. We will need this is the following chapter.

Lemma 5.21. *Let $\mathcal{C}$ be a skew circuit over $\mathbb{Z}[x]$ such that every input of the circuit is a polynomial of the form $\sum_{i=1}^{k} a_i x^i$ where the a_i are given in binary encoding. Then we can construct in logarithmic space a skew circuit $\mathcal{D}$ with inputs from $\{-1, 1, x\}$ such that $[\mathcal{C}] = [\mathcal{D}]$.*

Proof. Let $\mathcal{C}$ be a a skew circuit over $\mathbb{Z}[x]$ such that every input of the circuit is a polynomial of the form $\sum_{i=1}^{k} a_i x^i$ where the a_i are given in binary encoding. In a first step we split up every input sum into k addition gates: if A is a multiplication gate with $\mathsf{rhs}(A) = (\sum_{i=1}^{k} a_i x^i) \cdot B$ we set $\mathsf{rhs}(A) = \sum_{i=1}^{k} (a_i x^i \cdot B)$ where now the multiplication in every summand is meant as a multiplication gate and the sum is meant as a chain of addition gates in the circuit. For inputs of addition gates just split the sum directly. Now we can assume that every input is of the form ax^n where a is encoded binary and n is encoded unary. With iterated doubling as in the proof of Lemma 5.19 we can assume that $a = 1$ or $a = -1$. In a last step for inputs of the form x^n we can build multiplication chains of length n with input x to finally get the circuit $\mathcal{D}$. □

5.6 Circuit evaluation for polynomial rings

We have defined the circuit evaluation problem over a fixed algebraic structure $\mathcal{A}$, but sometimes we assume that a part of the algebraic structure is part of the input. For a unitary ring R and a set of variables $\{x_1, \ldots, x_k\}$ we consider circuit evaluation over the unitary polynomial ring $R[x_1, \ldots, x_k]$ where k is also part of the input. This problem is also known as polynomial identity testing over R ($\mathsf{PIT}(R)$). Recall that coRP is the class of complements of problems that can be solved in randomized polynomial time and coRNC is the class of complements of problems that are solvable by a randomized NC-circuit. For the rings $\mathbb{Z}$ and $\mathbb{Z}_p$ (p prime) the following result was shown in [45]; for $\mathbb{Z}_n$ with n composite, it was shown in [4].



(a) Skew Circuit $\mathcal{C}$ over $\mathbb{Z}[x]$

(b) Replacing x by X_{25}

(c) The resulting purely additive circuit

Figure 5.4: The transformation of a skew circuit over $(\mathbb{Z}[x], +, \cdot)$ into a purely additive circuit.

Theorem 5.22. *For each of the rings $\mathbb{Z}$ and $\mathbb{Z}_n$ ($n \geq 2$),* PIT *belongs to the class* coRP.

In [48] for PIT restricted to skew circuits the following result was shown:

Theorem 5.23. *For each of the rings $\mathbb{Z}$ and $\mathbb{Z}_n$ ($n \geq 2$),* PIT *restricted to skew circuits belongs to the class* coRNC.

Another important result in this context is the following:

Theorem 5.24 ([6])**.** *Polynomial identity testing for $\mathbb{Z}$ can be reduced to circuit evaluation for $(\mathbb{Z}, +\cdot)$.*

The proof of this theorem is based on the idea that we can replace the variables x_i by sufficient large integers α_i. This idea will be also used several times in this thesis.

We now define a class of circuits over unitary rings that in some sense lies between the class of standard circuits and the class of skew circuits:

Definition 5.25 (powerful skew circuit)**.** *Let R be a unitary ring. A circuit over the polynomial ring $R[x_1, \ldots, x_k]$ is powerful skew, if for every multiplication gate A one of its input is an input gate of the circuit and the input values are elements of the set $\{-1, 1\} \cup \{x_i^n \mid n \in \mathbb{N}, 1 \leq i \leq k\}$ where n is given in binary encoding.*

Notice that in the above definition the input set is not finite, but as for the max-plus semiring with binary encoded integers one could interpret the inputs of the form x_i^n as purely multiplicative subcircuits of size $\lceil \log n \rceil$ with inputs from $\{x_i \mid 1 \leq i \leq k\}$. We assume that powerful skew circuits are given in the form of the definition, since this simplifies the corresponding proofs and reductions a lot.

Note that the transformation of a powerful skew circuit over $R[x_1, \ldots, x_k]$ into an equivalent standard skew circuit requires an exponentially large blow-up. For instance, the smallest standard skew circuit for the polynomial x^n has size n, whereas x^n can be obtained by a powerful skew circuit as one gate of size $\lceil \log n \rceil$.

Recall that if k is a constant then skew circuits over the polynomial rings $\mathbb{Z}[x_1, \ldots, x_k]$ and $\mathbb{Z}_n[x_1, \ldots, x_k]$ can be evaluated in $\mathsf{C}_=\mathsf{L}$ (resp. DET), whereas for a non-fixed k the best upper bound is coRNC and no polynomial time algorithm is known. This is due to the fact that in the second case the number of monomials can be exponentially large in the size of the circuit. We will now show that this difference gets lost for powerful skew circuits.

A circuit over $R[x]$ is called a univariate circuit. Let $p(x_1, \ldots, x_k)$ be a polynomial and let $d \in \mathbb{N}$ such that $\deg(p, x_i) < d$ for all $1 \leq i \leq k$. Recall the definition of the univariate polynomial $\mathcal{U}_d(p)$ from Chapter 4:

$$\mathcal{U}_d(p) = p(y^1, y^d, \ldots, y^{d^{k-1}}).$$

The following lemma can be shown for arbitrary circuits, but we will only need it for powerful skew circuits.

Lemma 5.26. *Given a powerful skew circuit $\mathcal{C}$ for the polynomial $p(x_1, \ldots, x_k)$, the following can be computed in NC^2:*

(i) The binary encoding of a number d such that $\deg(p, x_i) < d$ for all $1 \leq i \leq k$ and

(ii) a powerful skew circuit $\mathcal{C}'$ that evaluates to $\mathcal{U}_d(p)$.

Proof. Let $\mathcal{C}$ be a powerful skew circuit for the polynomial $p(x_1, \ldots, x_k)$. In order to compute an upper bound on the degree $\deg(p, x_i)$, we construct a circuit over the max-plus semiring as follows: take the circuit $\mathcal{C}$. If A is an input gate that is labeled with x_i^n, then relabel A with the binary coded number n. Otherwise relabel A with 0. Moreover, for a gate A with $\mathsf{rhs}(A) = B + C$ (resp., $\mathsf{rhs}(A) = B \cdot C$) we set $\mathsf{rhs}(A) = \max(B, C)$ (resp., $\mathsf{rhs}(A) = B + C$). The resulting circuit is clearly skew. Therefore it can be evaluated in NC^2 by Lemma 5.18 and we can compute an upper bound for $\deg(p, x_i)$ for all $1 \leq i \leq k$ in parallel.

Once the number d is computed we simply replace every monomial x_i^n in the circuit $\mathcal{C}$ by y^N where $N = nd^{i-1}$. □

Note that the above reduction from multivariate to univariate circuits does not work for standard skew circuits: the output circuit will be powerful skew even if the input circuit is standard skew. For instance, the polynomial $\prod_{i=1}^{k} x_i$ (which can be produced by a standard skew circuit of size k) is transformed into the polynomial y^{2^k-1}, for which the smallest standard skew circuit has size $2^k - 1$.

In the following table the expressivenesses of the different types of circuits over $\mathbb{Z}[x]$ are shown. One can see that while the coefficients of polynomials generated by skew and powerful skew circuits with n gates are bounded by 2^n, standard circuit can generate polynomials with coefficients of size 2^{2^n}. On the other hand while powerful skew circuits can express polynomials of degree 2^n (as standard circuits can do), the degree of polynomials generated by skew circuits with n gates is bounded by n.

bounds	standard	powerful skew	skew
degree	2^n	2^n	n
coefficients	2^{2^n}	2^n	2^n

Definition 5.27 (powerful branching program)**.** *A* powerful branching program *is a branching program* (V, E, λ, s, t) *over a polynomial ring* $R[x_1, \ldots, x_k]$ *where every edge label* $\lambda(e)$ *is* $1, -1$ *or of the form* x^n *for some binary represented* $n \geq 1$*.*

As for skew circuits and branching programs over commutative unitary rings also the powerful versions are basically the same objects and can be transformed into each other in the same way. In Chapter 6 we show that polynomial identity testing for powerful skew circuits over the rings $\mathbb{Z}$ and $\mathbb{Z}_p$ for a prime p can be solved in coRNC.

5.7 Circuit evaluation for groups

A very well-studied setting of the circuit evaluation problem is circuit evaluation for finitely generated groups. Recall that the word problem was originally defined for groups. Since as mentioned above circuits can be seen as a compressed form of algebraic expressions, it makes sense that circuit evaluation for groups is also known as the compressed word problem. One might consider [57] for an extensive overview about this problem. For instance, it is shown there that for the following groups the circuit evaluation problem can be solved in polynomial time: finite groups, f.g. nilpotent groups, f.g. free groups, graph groups (also known as right-angled Artin groups or partially commutative groups), and virtually special groups, which are groups that have a finite index subgroup that embeds into a graph group. The latter groups form a rather large class that include for instance Coxeter groups, one-relator groups with torsion, fully residually free groups, and fundamental groups of hyperbolic 3-manifolds. For the exact definition of these groups one might also consider [57]. Some other exciting results are the following:

Theorem 5.28 ([57, Theorem 4.15])**.** *For every f.g. linear group the circuit evaluation problem belongs to the class* coRP*.*

This result is shown by reducing circuit evaluation for a f.g. linear group to $\mathsf{PIT}(\mathbb{Z})$ (in case the group consists of matrices over a field with characteristic 0) or $\mathsf{PIT}(\mathbb{Z}_n)$ (in case the group consists of matrices over a field with characteristic n) . Also a kind of converse of Theorem 5.28 is shown:

Theorem 5.29 ([57, Theorem 4.16])**.** *The problem* $\mathsf{CEP}(\mathsf{SL}_3(\mathbb{Z}))$ *and polynomial identity testing over* $\mathbb{Z}$ *are polynomial time reducible to each other.*

This result is shown by using a construction of Ben-Or and Cleve [20] for simulating arithmetic operations by matrix products.

In Chapter 7 we further investigate the relation between certain groups and some special cases of polynomial identity testing. We improve the polynomial time result for nilpotent groups to DET and show that for polycyclic groups circuit evaluation is as least as hard as $\mathsf{PIT}(\mathbb{Z})$ for powerful skew circuits w.r.t an NC^2-reduction. Furthermore we show the NC^2-equivalence between $\mathsf{PIT}(\mathbb{Z})$ (resp. $\mathsf{PIT}(\mathbb{Z}_n)$) for powerful skew circuits and circuit evaluation for the wreath product $\mathbb{Z} \wr \mathbb{Z}$ (resp. $\mathbb{Z}_n \wr \mathbb{Z}$). Subsequently we use these results to construct some more groups with a circuit evaluation problem in coRNC^2.

Chapter 6

Circuit evaluation for powerful skew circuits and equality testing for multi-dimensional SLPs

6.1 Introduction

Recall that for general circuits polynomial identity testing over $\mathbb{Z}$ and $\mathbb{Z}_n$ for $n \in \mathbb{N}$ is in coRP, while for skew circuits these problems are in coRNC. In this chapter we show that also for the more expressive powerful skew circuits polynomial identity testing over $\mathbb{Z}$ and $\mathbb{Z}_p$ ($= \mathbb{F}_p$ where $\mathbb{F}_p$ is the field of characteristic p) for a prime p can be solved in coRNC. The proof of this result has two main ingredients: the randomized identity testing algorithm of Agrawal and Biswas [4] and the modular polynomial powering algorithm of Fitch and Tompa [38]. After we have proved this result we show an application of it: a straight-line program is a context-free grammar that evaluates to a single word. A well-known problem is the question whether two straight-line programs evaluate to the same word. It is known that this problem can be solved in polynomial time, which was first shown independently in [44], [63] and [72]. The fastest known algorithm needs quadratic time under some assumptions on the machine model [47], but no efficient parallel algorithm is known yet. One can extent the concept of straight-line programs to the multi-dimensional case, where a straight-line program evaluates to a multi-dimensional picture. We show that the question whether two multi-dimensional straight-line programs evaluate to the same picture is equivalent to polynomial identity testing for powerful skew circuits over $\mathbb{Z}_2$. The results of this chapter have appeared in [2].

6.2 PIT for powerful skew circuits in coRNC

The main result of this section is the following:

Theorem 6.1. *For each of the rings $\mathbb{Z}$ and $\mathbb{Z}_p$ where p is a prime that can be part of the input in unary encoding,* PIT *for powerful skew circuits belongs to the class* coRNC^2.

Let us start with the identity testing algorithm of Agrawal and Biswas. We will only need the version for the polynomial ring $\mathbb{Z}_p[x]$ where p is a prime number.

Consider a polynomial $P(x) \in \mathbb{Z}_p[x]$ of degree d. The algorithm of Agrawal and Biswas consists of the following steps (later we will apply this algorithm to the polynomial defined by a powerful skew circuit), where $0 < \epsilon < 1$ is an error parameter:

1. Let ℓ be a number with $\ell \geq \log d$ and $t = \max\{\ell, \frac{1}{\epsilon}\}$

2. Find the smallest prime number r such that $r \neq p$ and r does not divide any of $p-1$, $p^2-1, \ldots, p^{\ell-1}-1$. It is argued in [4] that $r \in O(\ell^2 \log p)$.

3. Randomly choose a tuple $b = (b_0, \ldots, b_{\ell-1}) \in \{0,1\}^\ell$ and compute the polynomial $T_{r,b,t}(x) = Q_r(A_{b,t}(x))$ where $Q_r(x) = \sum_{i=0}^{r-1} x^i$ is the r^{th} cyclotomic polynomial and $A_{b,t} = x^t + \sum_{i=0}^{\ell-1} b_i \cdot x^i$.

4. Accept, if $P(x) \bmod T_{r,b,t} = 0$, otherwise reject.

Clearly, if $P(x) = 0$, then the above algorithm accepts with probability 1. For a non-zero polynomial $P(x)$, Agrawal and Biswas proved the following theorem:

Theorem 6.2 ([4])**.** *Let $P(x) \in \mathbb{Z}_p[x]$ be a non-zero polynomial of degree d. The above algorithm rejects $P(x)$ with probability at least $1-\varepsilon$.*

The second result we are using was shown by Fich and Tompa:

Theorem 6.3 ([38])**.** *The following computation can be done in NC^2:*
Input: A unary encoded prime number p, polynomials $a(x), q(x) \in \mathbb{F}_p[x]$ such that $\deg(a(x)) < \deg(q(x)) = d$, and a binary encoded number m.
Output: The polynomial $a(x)^m \bmod q(x)$.

In [38], it is stated that the problem can be solved using circuits of depth $(\log n)^2 \log\log n$ for the more general case that the underlying field is $\mathbb{F}_{p^\ell}$ where p and ℓ are given in unary representation. The main bottleneck is the computation of an iterated matrix product $A_1 A_2 \cdots A_k$ (for k polynomial in n) of $(d \times d)$-matrices over the field $\mathbb{F}_{p^\ell}$. In our situation (where the field is $\mathbb{F}_p$) we easily obtain an NC^2-algorithm for this step: two $(d \times d)$-matrices over $\mathbb{F}_p$ can be multiplied in $\mathsf{DLOGTIME}$-uniform TC^0. Then we compute the product $A_1 A_2 \cdots A_k$ by a balanced binary tree of depth $\log k$.

Proof of Theorem 6.1. We first prove the theorem for the case of a powerful skew circuit $\mathcal{C}$ over the ring $\mathbb{Z}_p$ where the prime number p is part of the input but specified in unary notation.

Let p be a unary encoded prime number and $\mathcal{A} = (\{1, \ldots, n\}, 1, n, \lambda)$ be a powerful branching program that is equivalent to $\mathcal{C}$. Let $P(x) = [\mathcal{A}] \in \mathbb{Z}_p[x]$. Fix an error probability $0 < \varepsilon < 1$. Our randomized NC^2-algorithm is based on the identity testing algorithm of Agrawal and Biswas. It accepts with probability 1 if $P(x) = 0$ and accepts with probability at most ϵ if $P(x) \neq 0$. Let us go through the four steps of the Agrawal-Biswas algorithm to see that they can be implemented in NC^2.

Step 1. An upper bound on the degree of $P(x)$ can be computed by a skew circuit over the max-+-semiring with binary encoded inputs in NC^2 as in the proof of Lemma 5.26. For the number ℓ we can take the number of bits of this degree bound, which is polynomial in the input size.

Step 2. For the prime number r we know that $r \in O(\ell^2 \log p)$, which is a polynomial bound. Hence, we can test in parallel all possible candidates for r. For a certain candidate r, we check in parallel whether it is prime (recall that r is of polynomial size) and whether it divides any of the numbers $p-1$, $p^2-1, \ldots, p^{\ell-1}-1$. The whole computation is possible in NC^1.

Step 3. Let $b = (b_0, \ldots, b_{\ell-1}) \in \{0,1\}^\ell$ be the chosen tuple. We have to compute the polynomial $T_{r,b,t}(x) = Q_r(A_{b,t}(x))$ where $Q_r(x) = \sum_{i=0}^{r-1} x^i$ and $A_{b,t} = x^t + \sum_{i=0}^{\ell-1} b_i \cdot x^i$. This is an instance of iterated multiplication (for the powers $A_{b,t}(x)^i$) and iterated addition of polynomials. Hence, by Theorem 2.1 this step can be carried out in $\mathsf{DLOGTIME}$-uniform TC^0. Note that the degree of $T_{r,b,t}(x)$ is $\ell \cdot (r-1) \in O(\ell^3 \log p)$, i.e., polynomial in the input size.

Step 4. For the last step, we have to compute $P(x) \bmod T_{r,b,t}(x)$. For this, we consider in parallel all monomials x^k that occur in an edge label of our powerful algebraic branching program $\mathcal{A}$. Recall that k is given in binary notation. Using the Fich-Tompa algorithm we compute $x^k \bmod$

$T_{r,b,t}(x)$ (with $a(x) = x$) in NC^2. We then replace the edge label x^k by $(x^k \bmod T_{r,b,t}(x))$. Let $\mathcal{B}$ be the resulting algebraic branching program. Every polynomial that appears as an edge label in $\mathcal{B}$ is now given in the form of Lemma 5.21 for skew circuits. Hence, by Lemma 5.19 we can compute in DET the output polynomial $[\mathcal{B}]$. Clearly, $P(x) \bmod T_{r,b,t}(x) = [\mathcal{B}] \bmod T_{r,b,t}(x)$. The latter polynomial can be computed in TC^0 by Theorem 2.1.

Let us now prove Theorem 6.1 for the ring $\mathbb{Z}$. Let $\mathcal{A} = (\{1, \ldots, n\}, 1, n, \lambda)$ be a powerful branching program over $\mathbb{Z}$ with n nodes and let $P(x) = [\mathcal{A}]$. Let us first look at the coefficients of $P(x)$. Since there are at most 2^n many paths from s to t in $\mathcal{A}$, every coefficient of the polynomial $P(x)$ belongs to the interval $[-2^n, 2^n]$, so $P(x) = 0$ if and only if $P(x) \equiv 0 \bmod a$ for an $a \geq 2^{n+1}$. Let $p_1, \ldots, p_{n+1}$ be the first $n+1$ prime numbers. Then $\prod_{i=1}^{n+1} p_i > 2^{n+1}$. Each prime p_i is polynomially bounded in n and the list of primes can be computed in logarithmic space, see e.g. [27].

The Chinese remainder theorem (see e.g. [92]) implies that $P(x) \equiv 0 \bmod p_i$ for all $1 \leq i \leq n+1$ if and only if $P(x) \equiv 0 \bmod \prod_{i=1}^{n+1}$ what is equivalent to $P(x) = 0$. We can carry out the former tests in parallel using the above algorithm for a unary encoded prime number. The overall algorithm accepts if we accept for every prime p_i. If $P(x) = 0$, then we will accept for every $1 \leq i \leq n+1$ with probability 1, hence the overall algorithm accepts with probability 1. On the other hand, if $P(x) \neq 0$, then there exists a prime p_i $(1 \leq i \leq n+1)$ such that the algorithm rejects with probability at least $1 - \varepsilon$. Hence, the overall algorithm will reject with probability at least $1 - \varepsilon$ as well. □

6.3 Multi-dimensional straight-line programs

In this section we use Theorem 6.1 to show that the equivalence of two straight-line programs (SLPs) can be decided in coRNC. An SLP is a context-free grammar that generates exactly one word over the alphabet Γ. It can also be seen as a circuit over the free monoid generated by Γ, but the first point of view seems to be more natural. On the other hand every circuit $\mathcal{C}$ over a monoid generated by some set Γ corresponds to an SLP $\mathcal{G}(\mathcal{C})$ by interpreting multiplication gates as concatenation of words over the alphabet Γ. To proof our result in a slight extended setting, we first extend the concept of SLPs to multi-dimensional straight-line programs:

Definition 6.4 (n-dimensional picture)**.** *Let Γ be a finite alphabet. For $l \in \mathbb{N}$ let $[0, l] = \{0, 1, \ldots, l\}$. An n-dimensional picture over Γ is a mapping $p : \prod_{j=1}^{n} [0, l_j - 1] \to \Gamma$ for some $l_j \geq 1$.*

Let $\mathrm{dom}(p) = \prod_{j=1}^{n} [0, l_j - 1]$. For $1 \leq j \leq n$ we define $|p|_j = l_j$ as the length of p in the j-th dimension. Note that one-dimensional pictures are simply finite words. Let Γ_n^* denote the set of n-dimensional pictures over Γ. On this set we can define partially defined concatenation operations $\circ_i$ $(1 \leq i \leq n)$ as follows: for pictures $p, q \in \Gamma_n^*$, the picture $p \circ_i q$ is defined if and only if $|p|_j = |q|_j$ for all $1 \leq j \leq n$ with $i \neq j$. In this case, we have $|p \circ_i q|_j = |p|_j \, (= |q|_j)$ for $j \neq i$ and $|p \circ_i q|_i = |p|_i + |q|_i$. Let $l_j = |p \circ_i q|_j$. For a tuple $(k_1, \ldots, k_n) \in \prod_{j=1}^{n} [0, l_j - 1]$ we finally set

$$(p \circ_i q)(k_1, \ldots, k_n) = \begin{cases} p(k_1, \ldots, k_n) \text{ if } k_i < |p|_i \\ q(k_1, \ldots, k_{i-1}, k_i - |p|_i, k_{i+1}, \ldots, k_n) \text{ if } k_i \geq |p|_i \end{cases} .$$

These operations generalize the concatenation of finite words.

Example 6.5. [1]

$$\left(\begin{array}{|ccc|}\hline 1&0&0\\0&1&1\\\hline\end{array}\ \circ_2\ \begin{array}{|ccc|}\hline 1&0&1\\\hline\end{array}\right)\circ_1\left(\begin{array}{|cc|}\hline 1&0\\0&1\\\hline\end{array}\ \circ_2\ \begin{array}{|cc|}\hline 1&1\\\hline\end{array}\right)$$

$$=\quad \begin{array}{|ccc|}\hline 1&0&1\\1&0&0\\0&1&1\\\hline\end{array}\ \circ_1\ \begin{array}{|cc|}\hline 1&1\\1&0\\0&1\\\hline\end{array}\ =\ \begin{array}{|ccccc|}\hline 1&0&1&1&1\\1&0&0&1&0\\0&1&1&0&1\\\hline\end{array}$$

Definition 6.6 (n-dimensional straight-line program)**.** *An n-dimensional straight-line program over the terminal alphabet Γ is a triple $\mathbb{A} = (V, S, \mathsf{rhs})$ where V is a finite set of variables, $S \in V$ is the start variable, and rhs maps each variable A to its right-hand side $\mathsf{rhs}(A)$, which is either a terminal symbol $a \in \Gamma$ or an expression of the form $B \circ_i C$ where $B, C \in V$ and $1 \leq i \leq n$ such that the following additional conditions are satisfied:*

- *The relation $\{(A, B) \in V \times V \mid A \text{ occurs in } \mathsf{rhs}(B)\}$ is acyclic.*
- *Define $|A|_i$ for $A \in V$ and $1 \leq i \leq n$ as follows: if $\mathsf{rhs}(A) \in \Gamma$ then $|A|_i = 1$ for all i. If $\mathsf{rhs}(A) = B \circ_i C$ then $|A|_i = |B|_i + |C|_i$. We require that $|B|_j = |C_j|$ for $j \neq i$ and set $|A|_j = |B|_j$.*

These conditions ensure that every variable A evaluates to a unique n-dimensional picture $\mathsf{val}_{\mathbb{A}}(A)$ such that $|\mathsf{val}_{\mathbb{A}}(A)|_i = |A|_i$ for all $1 \leq i \leq n$. Finally, $\mathsf{val}(\mathbb{A}) = \mathsf{val}_{\mathbb{A}}(S)$ is the picture defined by $\mathbb{A}$. We omit the index $\mathbb{A}$ if the underlying SLP is clear from the context. We define the size of the SLP $\mathbb{A} = (V, S, \mathsf{rhs})$ as $|\mathbb{A}| = |V|$.

For all dimensions i it is straightforward to define an SLP $\mathbb{A}$ of size m such that $|\mathsf{val}(\mathbb{A})|_i = 2^m$. Hence, an SLP can be seen as a compressed representation of the picture it generates, and an exponential compression ratio can be achieved in this way.

6.4 Equality testing for compressed strings and n-dimensional pictures

Given two n-dimensional SLPs we want to know whether they evaluate to the same picture. In [21] it was shown that this problem belongs to coRP by translating it to polynomial identity testing for $\mathbb{Z}_2$. For a given n-dimensional picture $p : \mathrm{dom}(p) \to \{0, 1\}$ we define the polynomial

$$f_p(x_1, ..., x_n) = \sum_{(e_1,...,e_n)\in \mathrm{dom}(p)} p(e_1, ..., e_n) \prod_{i=1}^{n} x_i^{e_i}.$$

We consider f_p as a polynomial from $\mathbb{Z}_2[x_1, \ldots, x_n]$. For two n-dimensional pictures p and q such that $|p|_i = |q|_i$ for all $1 \leq i \leq n$ we clearly have $p = q$ if and only if $f_p + f_q = 0$ (recall that coefficients are from $\mathbb{Z}_2$). In [21], it was observed that from an SLP $\mathbb{A}$ for a picture P, one can easily construct a circuit over $\mathbb{Z}_2[x_1, \ldots, x_n]$ for the polynomial f_p, which leads to a coRP-algorithm for equality testing. Since the circuit for f_p is actually powerful skew, we get the following result:

Theorem 6.7. *The question whether two n-dimensional SLPs $\mathbb{A}$ and $\mathbb{B}$ evaluate to the same n-dimensional picture is in coRNC^2 (here, n is part of the input given in unary encoding).*

Proof. Let $\mathbb{A}_1 = (V_1, \mathsf{rhs}_1, S_1)$ and $\mathbb{A}_2 = (V_2, \mathsf{rhs}_2, S_2)$ be n-dimensional SLPs over the alphabet Γ. We can assume that $V_1 \cap V_2 = \emptyset$ and $\Gamma = \{0, 1\}$ (if $\Gamma = \{a_1, \ldots, a_k\}$ then we encode a_i by $0^i 1^{k-i}$).

First we calculate $|A|_i$ for every $1 \leq i \leq n$ and every $A \in V_1 \cup V_2$ in DET by evaluating additive circuits over $\mathbb{N}$ (by Lemma 5.11). If $|S_1|_i \neq |S_2|_i$ for at least one $1 \leq i \leq n$, then we have $\mathsf{val}(\mathbb{A}_1) \neq \mathsf{val}(\mathbb{A}_2)$. Otherwise, we construct the circuit

$$\mathcal{C} = (V_1 \cup V_2 \cup \{S\}, \mathsf{rhs}, S)$$

[1] Notice that in contrast to matrices, the coordinates of a picture increase from bottom to top

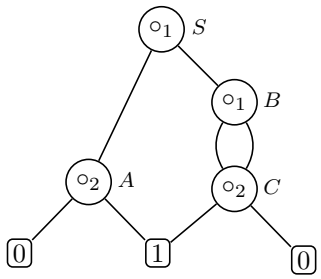
over $\mathbb{Z}_2[x_1, \ldots, x_n]$ with:

$$\mathsf{rhs}(A) = B + x_k^{|B|_k} \cdot C \text{ if } \mathsf{rhs}_1(A) = B \circ_k C \text{ or } \mathsf{rhs}_2(A) = B \circ_k C,$$
$$\mathsf{rhs}(A) = a \text{ if } \mathsf{rhs}_1(A) = a \in \{0,1\} \text{ or } \mathsf{rhs}_2(A) = a \in \{0,1\}, \text{ and}$$
$$\mathsf{rhs}(S) = S_1 + S_2$$

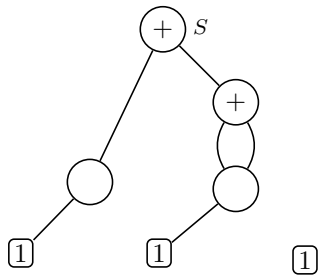
Then $[\mathcal{C}] = f_{\mathsf{val}(\mathbb{A}_1)} + f_{\mathsf{val}(\mathbb{A}_2)}$ and so $[\mathcal{C}] = 0$ if and only if $\mathsf{val}(\mathbb{A}_1) = \mathsf{val}(\mathbb{A}_2)$. After replacing 0 by $1 - 1$ and splitting right-hand sides of the form $B + x^n \cdot C$ the circuit $\mathcal{C}$ becomes a powerful skew circuit. Hence, Theorem 6.1 allows to check in coRNC^2 whether $[\mathcal{C}] = 0$. □

It should be noted that even in the one-dimensional case (where equality testing for SLPs can be done in polynomial time [44, 63, 72]), no randomized NC-algorithm was known before.

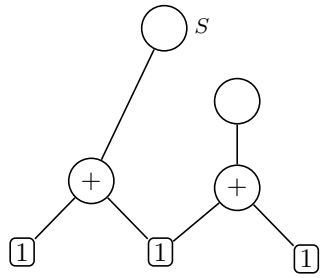
Example 6.8. *The SLP $\mathbb{A}$ in Figure 6.1 in part (a) evaluates to the picture* $\begin{array}{|ccc|}\hline 1 & 0 & 0 \\ 0 & 1 & 1 \\ \hline \end{array}$ *where dimension 1 is interpreted as the width and dimension 2 is interpreted as the height of the picture. The corresponding polynomial is $p = x_1 + x_1^2 + x_2$. In picture (b) and (c) there are the corresponding additive circuits that evaluate to the lengths of the picture in dimension 1 and dimension 2. The first one evaluates to 3 and the second one to 2. Finally in picture (d) the corresponding powerful skew circuit $\mathcal{C}$ is constructed as described in the proof of Theorem 6.7. Notice that $\mathcal{C}$ evaluates in fact to $x_1 + x_1^2 + x_2$.*



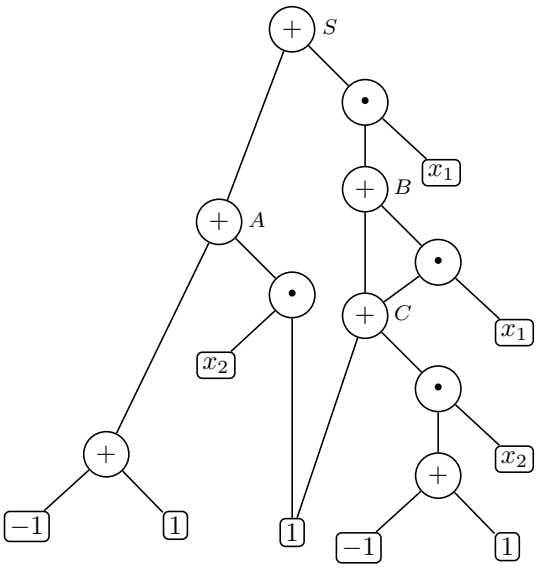
(a) SLP $\mathbb{A}$



(b) Additive circuit that calculates the length in dimension 1



(c) Additive circuit that calculates the length in dimension 2



(d) Corresponding Circuit $\mathcal{C}$

Figure 6.1: The transformation of a 2-dimensional SLP $\mathbb{A}$ into the corresponding powerful skew circuit $\mathcal{C}$.

Chapter 7

Circuit evaluation for groups

7.1 Introduction

As mentioned above the circuit evaluation problem for groups is also known as the compressed word problem. This is due to the fact that every circuit $\mathcal{C}$ over a group G with generating set Σ corresponds to an SLP $\mathbb{A}$ over the alphabet $\Sigma \cup \Sigma^{-1}$ such that the word $\mathsf{val}(\mathbb{A})$ evaluates to the element $[\mathcal{C}]$ in G. We will denote the corresponding SLP to the circuit $\mathcal{C}$ by $\mathcal{G}(\mathcal{C})$. In some of the following proofs we will make use of this SLP, since sometimes we need properties as the length of a word. We will consider the circuit evaluation problem in the following form: given a circuit $\mathcal{C}$ over a group G, decide whether $\mathcal{C}$ evaluates to the group identity. In the following section we show that circuit evaluation for the wreath product $\mathbb{Z} \wr \mathbb{Z}$ (resp., $\mathbb{Z}_n \wr \mathbb{Z}$) is NC^2-equivalent to polynomial identity testing over $\mathbb{Z}$ (resp., $\mathbb{Z}_n$) for powerful skew circuits. The idea behind this is to interpret the right part of the wreath product as exponents of variables and the left part as coefficients. It follows that circuit evaluation over $\mathbb{Z} \wr \mathbb{Z}$ and $\mathbb{Z}_p \wr \mathbb{Z}$ for a prime p can be solved in coRNC. After this we consider nilpotent groups. As mentioned in Section 3.4 every finitely generated nilpotent group has a subgroup of finite index that can be embedded in $\mathsf{UT}_d(\mathbb{Z})$ for some $d \in \mathbb{N}$. Then we use the idea from [57] that a circuit over a matrix group can be transformed into a circuit over the corresponding ring. This leads to a circuit over $(\mathbb{Z}, +, \cdot)$ with constant multiplication depth d for which circuit evaluation is in $\mathsf{C_=L}$ by Lemma 5.10. A related result was recently shown by Miasnikov and Weiß in [66], where they showed that the word problem for nilpotent groups where in the input word subwords of the form a^n for a generator a and a binary encoded n are allowed, is in $\mathsf{DLOGTIME}$-uniform TC^0. After this we take a closer look at circuit evaluation for the groups $\mathsf{UT}_d(\mathbb{Z})$ where d is part of the input. With the same techniques as before we construct a circuit over $(\mathbb{Z}, +, \cdot)$ with polynomially bounded formal degree. By Lemma 5.13 circuit evaluation for such circuits is in $\mathsf{C_=LogCFL}$. In the fourth section we mix up some of the previous results to show for wreath products of some certain groups that their circuit evaluation problem is in coRNC resp. in DET. In the last section we consider circuit evaluation for polycyclic groups. Recall that polycyclic groups are linear, and hence their circuit evaluation problem is in coRP. We show that there is a polycyclic group G such that polynomial identity testing over $\mathbb{Z}$ for powerful skew circuits is NC^2-reducible to its circuit evaluation problem. Again we use the idea to interpret the circuit over a matrix group as a circuit over the corresponding ring. It is still an open problem if there is a non-nilpotent polycyclic group with a circuit evaluation problem in coRNC. For a more extensive background about circuit evaluation for groups one might consult [54], [55], resp.[57]. The results of this chapter have appeared in [1] and [2].

7.2 Circuit evaluation for wreath products

As a second application of polynomial identity testing for powerful skew circuits we will consider the circuit evaluation problem for wreath products of finitely generated abelian groups.

In this section, we explore the relationship between circuit evaluation for the wreath product $\mathbb{Z} \wr \mathbb{Z}$ and $\mathsf{PIT}(\mathbb{Z})$ for powerful skew circuits. We show that these two problems are equivalent w.r.t. NC^2-reductions:

let $G = \mathbb{Z} \wr \mathbb{Z}$. Let $\Gamma = \{a, t, a^{-1}, t^{-1}\}$ be the generating set of G as described in Section 3.5. So, with a (resp., a^{-1}) we move the cursor to the right (resp., left) and with t (resp., t^{-1}) we add 1 (resp., subtract 1) from the value at the current cursor position.

For a word $w \in \Gamma^*$ we define $|w|_a$ as the number of occurrences of a in w and $\Delta(w) = |w|_a - |w|_{a^{-1}} \in \mathbb{Z}$. The word w is *positive* if $\Delta(u) \geq 0$ for every prefix u of w that ends with t or t^{-1}. The word w is *well-formed*, if it is positive and $\Delta(w) = 0$. If w is positive and $(f, g) \in G$ is a group element represented by the word w, then $f(x) \neq 0$ implies that $x \in \mathbb{N}$ (intuitively, the $\mathbb{Z}$-generator t or its inverse is never added to a position outside of $\mathbb{N}$). If in addition w is well-formed then $g = 0$. For a given positive word $w \in \Gamma^*$ we define a polynomial $p_w(x) \in \mathbb{Z}[x]$ inductively as follows:

- $p_\varepsilon(x) = 0$.
- If $w = ua$ or $w = ua^{-1}$, then $p_w(x) = p_u(x)$.
- If $w = ut^\delta$ with $\delta \in \{1, -1\}$, then $p_w(x) = p_u(x) + \delta \cdot x^d$ where $d = \Delta(w) = \Delta(u)$.

If the positive word w represents the group element $(f, g) \in G$, then the polynomial $p_w(x)$ encodes the mapping f in the following sense: the coefficient of the monomial x^e in $p_w(x)$ is exactly $f(e)$. In particular, the following equivalence holds for every positive word $w \in \Gamma^*$:

$$w = (0, 0) \text{ in } G \quad \Leftrightarrow \quad (p_w(x) = 0 \text{ and } \Delta(w) = 0)$$

where $(0, 0)$ is the group identity of G.

Lemma 7.1. *From a given circuit $\mathcal{C}$ over $\mathbb{Z} \wr \mathbb{Z}$ with inputs from the generating set Γ one can compute in NC^2 a powerful skew circuit $\mathcal{D}$ over $\mathbb{Z}[x]$ such that $[\mathcal{D}] = p_w(x)$ where $w = a^k\,[\mathcal{C}]\,a^{-k}$ and $k = |\mathsf{val}(\mathcal{G}(\mathcal{C}))|$. In particular, $[\mathcal{C}] = (0, 0)$ in G if and only if ($[\mathcal{D}] = 0$ and $\Delta(\mathsf{val}(\mathcal{G}(\mathcal{C}))) = 0$).*

Proof. Let $k = |\mathsf{val}(\mathcal{G}(\mathcal{C}))|$. Our construction is divided into the following two steps:

Step 1. Using iterated squaring, we add further gates A_k and A_k^{-1} to $\mathcal{C}$ such that $[A_k] = a^k$ and $[A_k^{-1}] = a^{-k}$. Then, we define the circuit $\mathcal{C}'$ by defining $\mathsf{rhs}_{\mathcal{C}'}(A) = A_k t^\delta A_k^{-1}$ for every gate A with $\mathsf{rhs}_{\mathcal{C}}(A) = t^\delta$ ($\delta \in \{-1, 1\}$). All other right-hand sides of $\mathcal{C}$ are left unchanged. Then, $[\mathcal{C}'] = a^k\,[\mathcal{C}]\,a^{-k}$.

Let $\mathcal{C}' = (V, S, \mathsf{rhs}_{\mathcal{C}'})$ for the further consideration. Note that for every $A \in V$, the word $\mathsf{val}_{\mathcal{G}(\mathcal{C}')}(A)$ is positive. Hence, for every $A \in V$ we can define the polynomial $p_A(x) := p_{\mathsf{val}_{\mathcal{G}(\mathcal{C}')}(A)}(x)$. Moreover, let $d_A = \Delta(\mathsf{val}_{\mathcal{G}(\mathcal{C}')}(A)) \in \mathbb{Z}$; these numbers d_A can be computed by an additive circuit in $\mathsf{DET}(\subseteq \mathsf{NC}^2)$.

For every $A \in V$ let

$$m_A = \min(\{\Delta(u) \mid u \text{ is a prefix of } \mathsf{val}_{\mathcal{G}(\mathcal{C}')}(A) \text{ that ends with } t \text{ or } t^{-1}\})$$

where we set $\min(\emptyset) = 0$. Since $\mathsf{val}_{\mathcal{G}(\mathcal{C}')}(A)$ is positive, we have $m_A \geq 0$. The polynomial $p_A(x)$ can be uniquely written as

$$p_A(x) = x^{m_A} \cdot q_A(x),$$

for a polynomial $q_A(x)$. The numbers m_A can be computed in NC^2, using the following identity where $\alpha(A)$ denotes the set of symbols occurring in $\mathsf{val}_{\mathcal{G}(\mathcal{C}')}(A)$.

$$m_A = \begin{cases} 0 & \text{if } \mathsf{rhs}_{\mathcal{C}'}(A) = a^\delta \\ k & \text{if } \mathsf{rhs}_{\mathcal{C}'}(A) = A_k t^\delta A_k^{-1} \\ \min\{m_B, d_B + m_C\} & \text{if } \mathsf{rhs}_{\mathcal{C}'}(A) = BC \text{ and } \alpha(C) \cap \{t, t^{-1}\} \neq \emptyset \neq \alpha(B) \cap \{t, t^{-1}\} \\ m_B & \text{if } \mathsf{rhs}_{\mathcal{C}'}(A) = BC \text{ and } \alpha(C) \cap \{t, t^{-1}\} = \emptyset \\ d_B + m_C & \text{if } \mathsf{rhs}_{\mathcal{C}'}(A) = BC \text{ and } \alpha(B) \cap \{t, t^{-1}\} = \emptyset \end{cases}$$

Note that these rules define a skew circuit with binary encoded inputs over the semiring $(\mathbb{Z} \cup \{\infty\}, \min, +)$. Hence, by Lemma 5.18 the circuit can be evaluated in NC^2.

Step 2. We now construct a circuit $\mathcal{D}$ such that for every $A \in V$ we have:

$$[A]_{\mathcal{D}} = q_A(x).$$

We define the rules of the circuit $\mathcal{D}$ as follows:

- If $\mathsf{rhs}_{\mathcal{C}'}(A) = a^\delta$ for $\delta \in \{-1, 1\}$, then we set $\mathsf{rhs}_{\mathcal{D}}(A) = -1 + 1$ [1].
- If $\mathsf{rhs}_{\mathcal{C}'}(A) = A_k t^\delta A_k^{-1}$ for $\delta \in \{-1, 1\}$, then we set $\mathsf{rhs}_{\mathcal{D}}(A) = \delta$.
- If $\mathsf{rhs}_{\mathcal{C}'}(A) = BC$ and $\alpha(C) \cap \{t, t^{-1}\} = \emptyset$, then we set $\mathsf{rhs}_{\mathcal{D}}(A) = B$.
- If $\mathsf{rhs}_{\mathcal{C}'}(A) = BC$ and $\alpha(B) \cap \{t, t^{-1}\} = \emptyset$, then we set $\mathsf{rhs}_{\mathcal{D}}(A) = C$.
- If $\mathsf{rhs}_{\mathcal{C}'}(A) = BC$ and $\alpha(B) \cap \{t, t^{-1}\} \neq \emptyset \neq \alpha(C) \cap \{t, t^{-1}\}$, then $m_A = \min\{m_B, d_B + m_C\}$ and we set $\mathsf{rhs}_{\mathcal{D}}(A) = (M_B \cdot B) + (M_C \cdot C)$ where

$$M_B = \begin{cases} 1 & \text{if } m_B \leq d_B + m_C \\ x^{m_B - d_B - m_C} & \text{if } m_B > d_B + m_C \end{cases}$$

$$M_C = \begin{cases} 1 & \text{if } m_B \geq d_B + m_C \\ x^{d_B + m_C - m_B} & \text{if } m_B < d_B + m_C. \end{cases}$$

Note that the resulting circuit is powerful skew and one can show by induction that $\mathcal{D}$ generates the desired polynomial. □

Corollary 7.2. *The circuit evaluation problem for $\mathbb{Z} \wr \mathbb{Z}$ is NC^2-reducible to $\mathsf{PIT}(\mathbb{Z})$ for powerful skew circuits.*

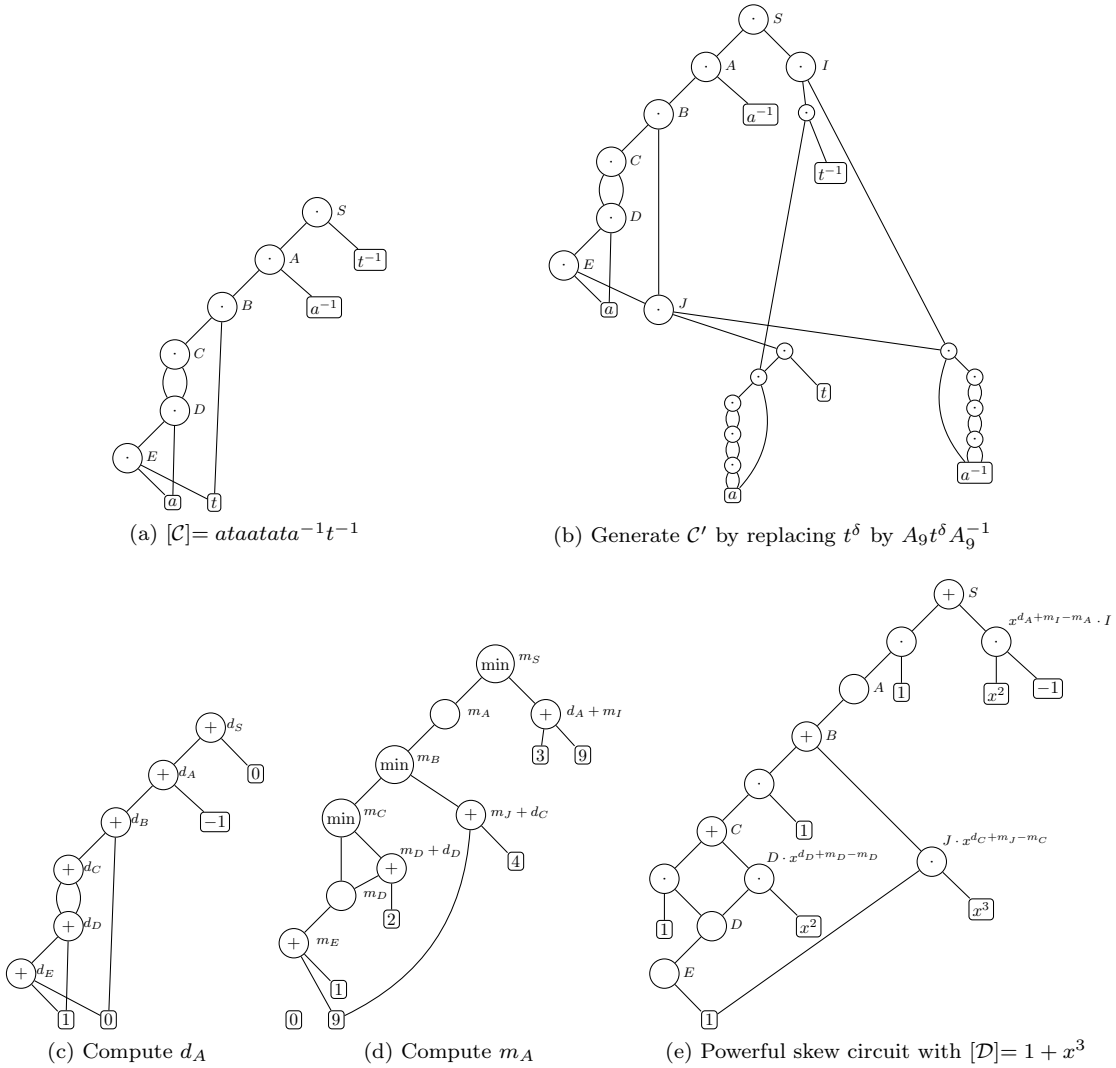
Example 7.3. *Figure 7.1 illustrates an example for the transformation of a circuit over $\mathbb{Z} \wr \mathbb{Z}$ that evaluates to the element $w = ataatata^{-1}t^{-1}$ from example 3.39 into a powerful skew circuit over $(\mathbb{Z}[x], +, \cdot)$. Since $|w| = 9$, in a first step we replace every input gate t (resp., t^{-1}) by the gate $A_9 t A_9^{-1}$ (resp., $A_9 t^{-1} A_9^{-1}$). The gates A_9 and A_9^{-1} evaluate to a^9 and a^{-9} by iterated squaring in picture (b). To construct the final circuit $\mathcal{D}$ we first need to calculate the values d_A and m_A for every gate $A \in V$. This is done by the additive circuit in picture (c) and the skew circuit over $(\mathbb{Z} \cup \{\infty\}, \min, +)$ in picture (d) that already uses the values d_A from the additive circuit. Finally in picture (e) we construct the powerful skew circuit over $(\mathbb{Z}[x], +, \cdot)$. Copy-gates are left unlabeled in the pictures. Notice that $p_{ataatata^{-1}t^{-1}}(x) = x + x^4$, the polynomial that corresponds to the word generated by $\mathcal{C}'$ is $x^{10} + x^{13}$ and $\mathcal{D}$ evaluates to this polynomial divided by $x^{m_S} = x^{10}$, therefore $[\mathcal{D}] = 1 + x^3$.*

In the rest of this section we show that $\mathsf{PIT}(\mathbb{Z})$ for powerful skew circuits can be reduced in NC^2 to $\mathsf{CEP}(\mathbb{Z} \wr \mathbb{Z})$. We need the following two lemmata which follow immediately from the definition of the polynomial $p_w(x)$:

Lemma 7.4. *Let $u, v \in \Gamma^*$ be well-formed. Then $w = uv$ is well-formed too and $p_w(x) = p_u(x) + p_v(x)$.*

Lemma 7.5. *Let $u \in \Gamma^*$ be well-formed, $n \in \mathbb{N}$ and let $w = a^n u a^{-n}$. Then w is well-formed too and $p_w(x) = x^n \cdot p_u(x)$.*

[1] Note that we excluded 0 from the input set of a powerful skew circuit for technical reasons.



(a) $[\mathcal{C}] = ataatata^{-1}t^{-1}$

(b) Generate $\mathcal{C}'$ by replacing t^δ by $A_9 t^\delta A_9^{-1}$

(c) Compute d_A

(d) Compute m_A

(e) Powerful skew circuit with $[\mathcal{D}] = 1 + x^3$

Figure 7.1: An illustration of the proof of Lemma 7.1.

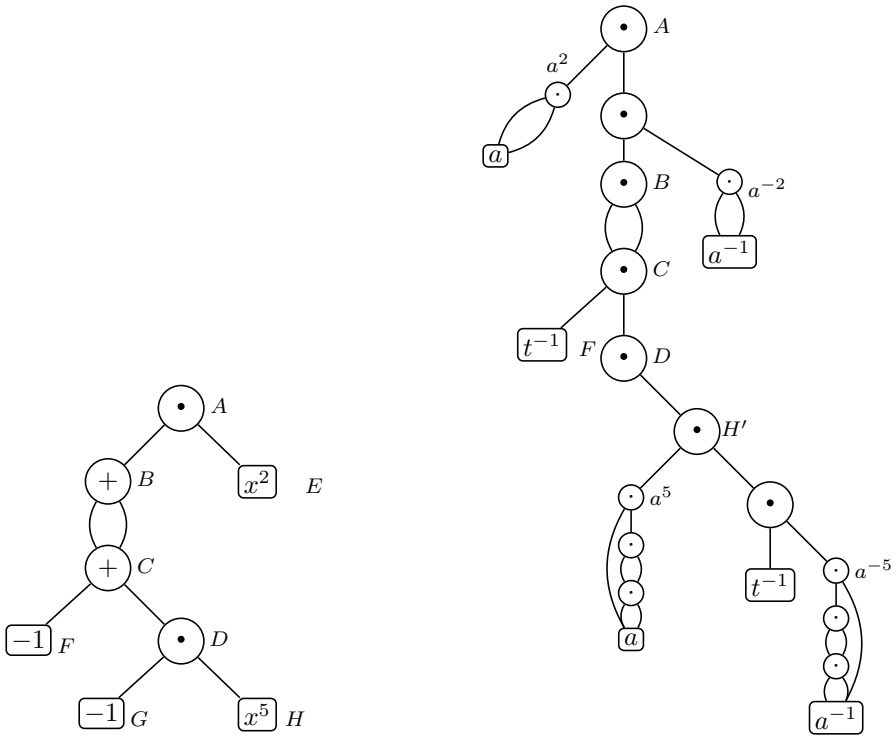
Now we can show the following result:

Lemma 7.6. *From a given powerful skew circuit $\mathcal{D}$ over the ring $\mathbb{Z}[x]$, one can compute in logarithmic space a circuit $\mathcal{C}$ with inputs from the generating set Γ such that the following holds:*

- $\mathsf{val}(\mathcal{G}(\mathcal{C}))$ *is well-formed and*
- $p_{\mathsf{val}(\mathcal{G}(\mathcal{C}))}(x) = [\mathcal{D}]$.

Proof. Let $\mathcal{D} = (V, \mathsf{rhs}_\mathcal{D}, S)$ be a powerful skew circuit over $\mathbb{Z}[x]$. The set of gates of $\mathcal{C}$ contains V, a disjoint copy $V' = \{A' \mid A \in V\}$ of V, and some auxiliary gates. The output gate is S. For every gate $A \in V$ we will have $p_A(x) = [A]_\mathcal{D}$ and for every gate $A' \in V'$ we will have $p_{A'}(x) = -[A]_\mathcal{D}$. We define the right-hand sides of the gates of $\mathcal{C}$ as follows:

- If $\mathsf{rhs}_\mathcal{D}(A) = x^n$ then we set $\mathsf{rhs}_\mathcal{C}(A) = a^n t a^{-n}$ and $\mathsf{rhs}_\mathcal{C}(A') = a^n t^{-1} a^{-n}$.
- If $\mathsf{rhs}_\mathcal{D}(A) = b$ with $b \in \{-1, 1\}$, then we set $\mathsf{rhs}_\mathcal{C}(A) = t^b$ and $\mathsf{rhs}_\mathcal{C}(A') = t^{-b}$.
- If $\mathsf{rhs}_\mathcal{D}(A) = B + C$, then we set $\mathsf{rhs}_\mathcal{C}(A) = BC$ and $\mathsf{rhs}_\mathcal{C}(A') = B'C'$. The correctness of this step follows from Lemma 7.4.



(a) Skew circuit $\mathcal{D}$ (b) Corresponding circuit over $\mathbb{Z} \wr \mathbb{Z}$

Figure 7.2: An illustration of the proof of Lemma 7.6.

- If $\mathsf{rhs}_{\mathcal{D}}(A) = B \cdot 1$, then we set $\mathsf{rhs}_{\mathcal{C}}(A) = B$ and $\mathsf{rhs}_{\mathcal{C}}(A') = B'$.
- If $\mathsf{rhs}_{\mathcal{D}}(A) = B \cdot -1$, then we set $\mathsf{rhs}_{\mathcal{C}}(A) = B'$ and $\mathsf{rhs}_{\mathcal{C}}(A') = B$.
- If $\mathsf{rhs}_{\mathcal{D}}(A) = B \cdot x^n$, then we set $\mathsf{rhs}_{\mathcal{C}}(A) = a^n B a^{-n}$ and $\mathsf{rhs}_{\mathcal{C}}(A') = a^n B' a^{-n}$. The correctness of this step follows from Lemma 7.5.

It follows by a straightforward induction that for every $A \in V$, the strings $\mathsf{val}_{\mathcal{G}(\mathcal{C})}(A)$ and $\mathsf{val}_{\mathcal{G}(\mathcal{C})}(A')$ are well-formed and that $P_S(x) = [\mathcal{D}]$. □

Example 7.7. *Figure 7.2 is an example for the transformation of a powerful skew circuit $\mathcal{D}$ into a circuit over $\mathbb{Z} \wr \mathbb{Z}$ from Lemma 7.6. The circuit $\mathcal{D}$ evaluates to the polynomial $(-2-2x^5)\cdot(x^2) = -2x^2 - 2x^7$. The right-hand side of A in the circuit $\mathcal{C}$ is set to a^2Ba^{-2}, where the left and the right part of this expression is generated by (iterated) squaring. The small gates in the figure are those that are only needed to generate exponential gates like a^5. Gate C evaluates to $t^{-1}a^5t^{-1}a^{-5}$ and the whole circuit $\mathcal{C}$ evaluates to $a^2(t^{-1}a^5t^{-1}a^{-5})^2a^{-2}$. So first the curser moves to position 2 and then repeats two time decreasing the values at position 2 and position 7. Then it goes back to position 0. So this element is indeed the one that corresponds to the polynomial $[\mathcal{D}]$.*

From Lemma 7.1 and Lemma 7.6 we directly obtain the following result:

Corollary 7.8. *The circuit evaluation problem for $\mathbb{Z} \wr \mathbb{Z}$ is equivalent w.r.t. NC^2-reductions to $\mathsf{PIT}(\mathbb{Z})$ for powerful skew circuits.*

In exactly the same way we can show this result for $\mathbb{Z}_n$ instead of $\mathbb{Z}$:

Corollary 7.9. *The circuit evaluation problem for $\mathbb{Z}_n \wr \mathbb{Z}$ ($n \geq 2$) is equivalent w.r.t. NC^2-reductions to $\mathsf{PIT}(\mathbb{Z}_n)$ for powerful skew circuits.*

7.3 Circuit evaluation for nilpotent groups

To consider circuit evaluation for finitely generated nilpotent groups we use two main properties of these groups: recall that every f.g. nilpotent group has a normal torsion-free f.g. nilpotent subgroup of finite index (Theorem 3.21) and that every torsion-free f.g. nilpotent group embeds into $\mathsf{UT}_d(\mathbb{Z})$ for some $d \in \mathbb{N}$ (Theorem 3.34). We use the latter fact to show the main result of this section:

Theorem 7.10. *Let $G \neq 1$ be a f.g. torsion-free nilpotent group. Then $\mathsf{CEP}(G)$ is complete for the class $\mathsf{C}_=\mathsf{L}$.*

For the lower bound let G be a non-trivial f.g. torsion-free nilpotent group. Since $G \neq 1$, G contains $\mathbb{Z}$. Since, as remarked in Section 5.3, $\mathsf{CEP}(\mathbb{Z})$ is hard for $\mathsf{C}_=\mathsf{L}$, the lower bound is clear. For the upper bound in Theorem 7.10, we show the following result:

Lemma 7.11. *For every $d \geq 1$, $\mathsf{CEP}(\mathsf{UT}_d(\mathbb{Z}))$ belongs to $\mathsf{C}_=\mathsf{L}$.*

For the rest of this section let us fix a number $d \geq 1$ and consider the unitriangular matrix group $\mathsf{UT}_d(\mathbb{Z})$. Consider a circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ with inputs from $\Gamma_d \cup \Gamma_d^{-1}$ where $\Gamma_d = \{T_{i,i+1} \mid 1 \leq i < d\}$ is the finite generating set of $\mathsf{UT}_d(\mathbb{Z})$ from Section 3.4.

We transform our circuit $\mathcal{C}$ in logarithmic space into a circuit $\mathcal{D}$ over $(\mathbb{Z}, +, \cdot)$ of multiplication depth at most d such that $\mathcal{C}$ evaluates to the identity matrix if and only if $\mathcal{D}$ evaluates to 0. Moreover, we also compute a structure-preserving partition of the multiplication gates of $\mathcal{D}$.

The degree bound in the following lemma will be only needed in Section 7.4.

Lemma 7.12. *From the circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over $\mathsf{UT}_d(\mathbb{Z})$ we can compute in logarithmic space a circuit $\mathcal{D}$ over $(\mathbb{Z}, +, \cdot)$ with $\mathsf{mdepth}(\mathcal{D}) \leq d$ and $\deg(\mathcal{D}) \leq 2(d-1)$ such that $[\mathcal{C}] = \mathsf{Id}$ if and only if $[\mathcal{D}] = 0$. In addition we can compute in logarithmic space a structure-preserving partition $\biguplus_{i=1}^{d} V_i$ of the set of all multiplication gates of $\mathcal{D}$.*

Proof. The set of gates of the circuit $\mathcal{D}$ is

$$W = \{A_{i,j} \mid A \in V, 1 \leq i < j \leq d\} \cup \{T\}$$

where T is the output gate. The idea is simple: gate $A_{i,j}$ will evaluate to the matrix entry $[A]_{\mathcal{C}}[i,j]$. To achieve this, we define the right-hand side mapping of the circuit $\mathcal{D}$ (which we denote again with rhs) as follows:

$$\mathsf{rhs}(A_{i,j}) = \begin{cases} M[i,j] & \text{if } \mathsf{rhs}(A) = M \in \Gamma_d \cup \Gamma_d^{-1} \\ B_{i,j} + C_{i,j} + \sum_{i<k<j} B_{i,k} \cdot C_{k,j} & \text{if } \mathsf{rhs}(A) = BC \end{cases}$$

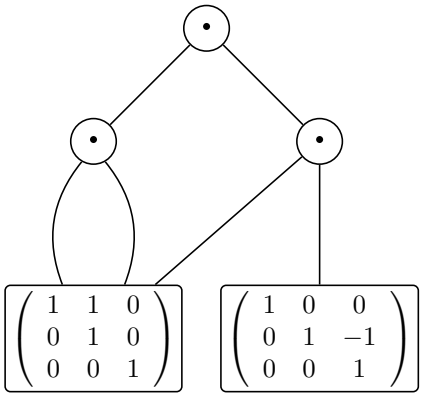
In the first line one has to notice that $M[i,j]$ is one of the numbers $-1, 0, 1$. The second line is simply the rule for matrix multiplication ($A_{i,j} = \sum_{k=1}^{d} B_{i,k}C_{k,j}$) taking into account that all matrices are unitriangular.

Now, $[\mathcal{C}]$ is the identity matrix if and only if all matrix entries $[S]_{\mathcal{C}}[i,j]$ ($1 \leq i < j \leq d$) are zero. But this is the case if and only if the sum of squares $\sum_{1 \leq i < j \leq d} [S]_{\mathcal{C}}[i,j]^2$ is zero. Hence, we finally define

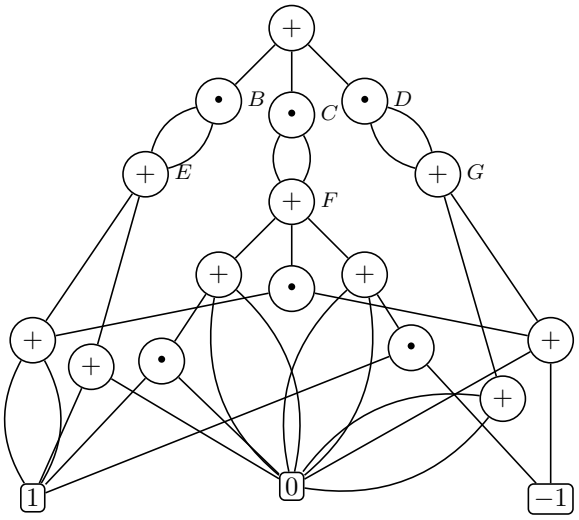
$$\mathsf{rhs}(T) = \sum_{1 \leq i < j \leq d} S_{i,j}^2.$$

Concerning the multiplication depth, note that the multiplication depth of the gate $A_{i,j}$ is bounded by $j-i$: the only multiplications in $\mathsf{rhs}(A_{i,j})$ are of the form $B_{i,k}C_{k,j}$ (and these multiplications are not nested). Hence, by induction, the multiplication depth of $A_{i,j}$ is bounded by $1+\max\{k-i, j-k \mid i < k < j\} = j - i$. It follows that every gate $S_{i,j}$ has multiplication depth at most $d-1$, which implies that the output gate T has multiplication depth at most d.

Similarly, it can be shown by induction that $\mathsf{deg}(A_{i,j}) \leq j - i$. Hence, $\mathsf{deg}(A_{i,j}) \leq d-1$ for all $1 \leq i < j \leq d$, which implies that the formal degree of the circuit is bounded by $2(d-1)$.



(a) Circuit $\mathcal{C}$ over $\mathsf{UT}_3(\mathbb{Z})$



(b) Circuit $\mathcal{D}$ of multiplication depth 2

Figure 7.3: Transformation of a circuit over $\mathsf{UT}_3(\mathbb{Z})$ into a circuit over $(\mathbb{Z}, +, \cdot)$.

The structure-preserving partition $\biguplus_{i=1}^{d} V_i$ of the set of all multiplication gates of $\mathcal{C}$ can be defined as follows: all gates corresponding to multiplications $B_{i,k} \cdot C_{k,j}$ in $\mathsf{rhs}(A_{i,j})$ are put into the set V_{j-i}. Finally, all gates corresponding to multiplications $S_{i,j}^2$ in $\mathsf{rhs}(T)$ are put into V_d. It is obvious that this partition is structure-preserving. □

Now Lemma 5.10 concludes the proof that $\mathsf{CWP}(\mathsf{UT}_d(\mathbb{Z}))$ belongs to $\mathsf{C}_=\mathsf{L}$.

Example 7.13. *Figure 7.3 illustrates the transformation from a circuit $\mathcal{C}$ over $\mathsf{UT}_3(\mathbb{Z})$ into a circuit over $(\mathbb{Z}, +, \cdot)$ of multiplication depth 2. The three multiplication gates B, C and D are used to square (and subsequently add) the values of E, F and G that represent the matrix entries of $[\mathcal{C}]$. The rest of the circuit can be split in three parts. The left part beneath E that evaluates to $[\mathcal{C}][1, 2]$, the right part beneath G that evaluates to $[\mathcal{C}][2, 3]$ and the middle part that evaluates to $[\mathcal{C}][1, 3]$. Notice that multiplication gates only appear in the middle part and their inputs are always from the left or the right part or input gates. This property illustrates why the multiplication depth of the circuit is 2 and (more general) why the transformation of circuits over $\mathsf{UT}_d(\mathbb{Z})$ leads to circuits with multiplication depth at most $d - 1$.*

So far, we have restricted to torsion-free f.g. nilpotent groups. For general f.g. nilpotent groups, we use the fact that every f.g. nilpotent group contains a torsion-free normal f.g. nilpotent

subgroup of finite index (Theorem 3.21) in order to show that the circuit evaluation problem for every f.g. nilpotent group belongs to the complexity class DET. To do this, we need the following result:

Theorem 7.14. *Let G be a finitely generated group. For every normal subgroup H of G with a finite index, $\mathsf{CEP}(G)$ is AC^0-reducible to $\mathsf{CEP}(H)$ and $\mathsf{CEP}(G/H)$.*

Proof. To show the theorem, we adopt the proof of [57, Theorem 4.4], where the statement is shown for polynomial time many-one reducibility instead of AC^0-reducibility. Let G be a finitely generated group with the finite generating set Σ and let H be a normal subgroup of G of finite index. Let $\{Hg_1, \ldots, Hg_n\}$ be the set of cosets of H in G where $g_1 = 1$. Moreover, let $\phi : G \to G/H$ be the canonical homomorphism. Now let $\mathcal{C} = (V, S, \mathsf{rhs})$ be a circuit with inputs from $\Sigma \cup \Sigma^{-1}$. We have to construct an AC^0-circuit with oracle gates for $\mathsf{CEP}(H)$ and $\mathsf{CEP}(G/H)$ that checks whether $[\mathcal{C}] = 1$ in G.

Consider the set of triples

$$W = \left\{(g_i, A, g_j^{-1}) \mid A \in V, 1 \leq i, j \leq n, g_i[A]_{\mathcal{C}} g_j^{-1} \in H\right\}.$$

In a first step, we construct the set of all these triples using $n^2|V|$ parallel $\mathsf{CEP}(G/H)$-oracle gates. More precisely, we construct for all $A \in V, 1 \leq i, j \leq n$ a circuit $\mathcal{C}_{A,i,j}$ that evaluates to the group element $\phi(g_i[A]_{\mathcal{C}} g_j^{-1}) \in G/H$. For this, we take the circuit $\mathcal{C}$, replace every input gate by the corresponding coset and add a new output gate $S_{A,i,j}$ with the right-hand side $\phi(g_i)A\phi(g_j)^{-1}$. The circuit $\mathcal{C}_{A,i,j}$ can be constructed in AC^0, and we have $[\mathcal{C}_{A,i,j}] = 1$ in G/H if and only if $g_i[A]_{\mathcal{C}} g_j^{-1} \in H$.

Note that $[\mathcal{C}_{S,1,1}] = 1$ in G/H if and only if $[\mathcal{C}]$ represents an element of the subgroup H. Thus, if it turns out that $[\mathcal{C}_{S,1,1}] \neq 1$ in G/H, then the whole circuit does not evaluate to the group identity. Otherwise (i.e., in case $[\mathcal{C}] \in H$), we construct a circuit $\mathcal{D}$ with inputs from the set

$$\{g_i a g_j^{-1} \mid a \in \Sigma \cup \Sigma^{-1}, 1 \leq i, j \leq n, g_i a g_j^{-1} \in H\}$$

that can be precomputed. $\mathcal{D}$ will evaluate to the same group element as $\mathcal{C}$.

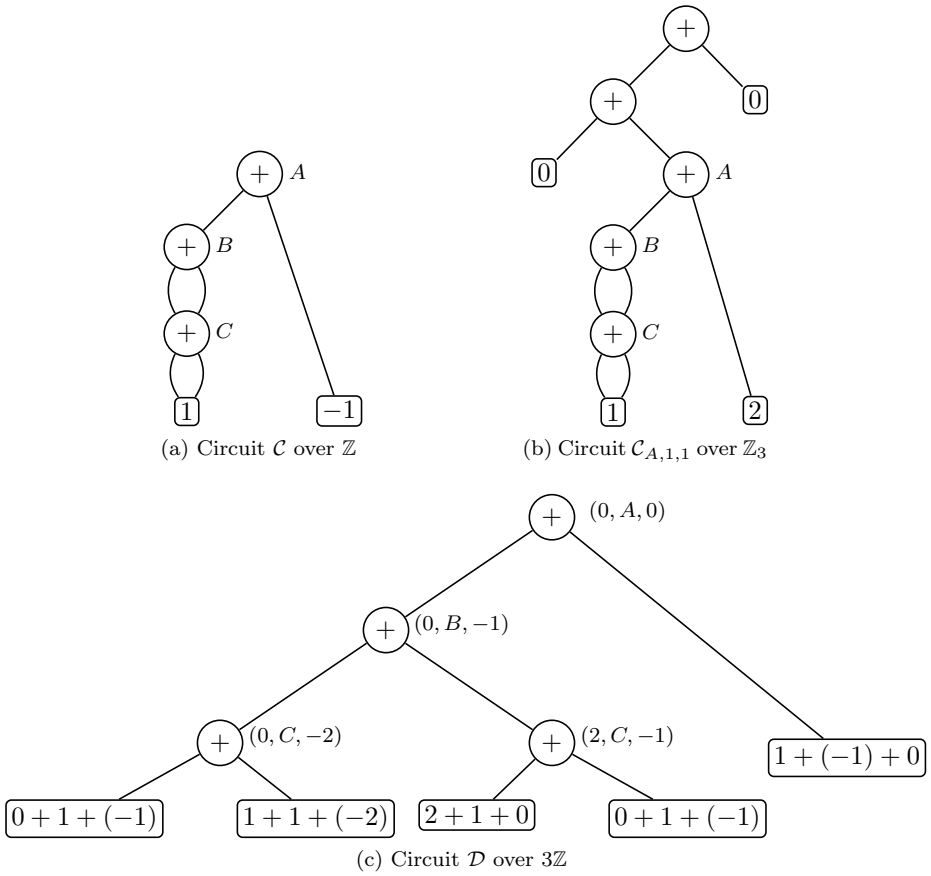
The set of gates of $\mathcal{D}$ is W, the output gate is (g_1, S, g_1^{-1}) and the right-hand sides are defined as follows: if $\mathsf{rhs}_{\mathcal{C}}(A) = a \in \Sigma \cup \Sigma^{-1}$, we set $\mathsf{rhs}_{\mathcal{D}}((g_i, A, g_j^{-1})) = g_i a g_j^{-1}$. If $\mathsf{rhs}_{\mathcal{C}}(A) = BC$ and $(g_i, A, g_j^{-1}) \in W$, then we determine the unique k so that $g_i[B]_{\mathcal{C}} g_k^{-1} \in H$. To do this, we have to go through the set W and look for the unique k such that $(g_i, B, g_k^{-1}) \in W$. Now we define $\mathsf{rhs}_{\mathcal{D}}((g_i, A, g_j^{-1})) = (g_i, B, g_k^{-1})(g_k, C, g_j^{-1})$. This construction can be carried out by an AC^0-circuit. Finally, it is straightforward to show that $[(g_i, A, g_j^{-1})]_{\mathcal{D}}$ represents the group element $g_i[A]_{\mathcal{C}} g_j^{-1} \in H$. Hence, we have $[\mathcal{C}] = 1$ in G, if and only if $[\mathcal{D}] = 1$ in H. This finishes our reduction. Note that the overall circuit consists of $n^2|V|$ parallel $\mathsf{CWP}(G/H)$-oracle gates followed by a single $\mathsf{CWP}(H)$-oracle gate.

□

Example 7.15. *Figure 7.4 illustrates the proof of Lemma 7.14 with the group $G = (\mathbb{Z}, +)$, the normal subgroup $H = (3\mathbb{Z}, +)$ and the finite quotient $G/H \cong (\mathbb{Z}_3, +)$. As usual the representing elements of the cosets are chosen as $g_1 = 0$, $g_2 = 1$ and $g_3 = 2$. In picture (a) there is a circuit $\mathcal{C}$ over $\mathbb{Z}$ that evaluates to 3, so in particular, $[\mathcal{C}] \in H$ which is tested by the circuit $\mathcal{C}_{A,1,1}$ in picture (b) over $\mathbb{Z}_3$. Finally in picture (c) there is the circuit $\mathcal{D}$ over H (notice that all inputs evaluate to elements of H) that evaluates to the same element as $\mathcal{C}$ does. For the construction of $\mathcal{D}$ we calculate the set $W = \{(a, A, b^{-1}) \mid A \in V, a, b, \in \{0, 1, 2\}, a[A]b^{-1} \bmod 3 = 0\}$ by the circuits $\mathcal{C}_{A,i,j}$ and take the needed gates from this set as described in the proof. The inverse of an element $a \in \mathbb{Z}_3$ is denoted by $-a$.*

We can now show the following main result of this section:

Theorem 7.16. *For every f.g. nilpotent group, the circuit evaluation problem is in DET.*



(a) Circuit $\mathcal{C}$ over $\mathbb{Z}$

(b) Circuit $\mathcal{C}_{A,1,1}$ over $\mathbb{Z}_3$

(c) Circuit $\mathcal{D}$ over $3\mathbb{Z}$

Figure 7.4: Illustration of the proof of Lemma 7.14.

Proof. Let G be a f.g. nilpotent group. If G is finite, then the result follows from Theorem 5.15 (every nilpotent group is solvable). If G is infinite, then by Theorem 3.21, G has a torsion-free normal subgroup H of finite index. By Theorem 3.20, H and G/H are nilpotent too; moreover H is finitely generated. By Theorem 7.10, $\mathsf{CEP}(H)$ belongs to $\mathsf{C}_=\mathsf{L} \subseteq \mathsf{DET}$. Moreover, by Theorem 5.15, $\mathsf{CEP}(G/H)$ belongs to DET as well. Finally, Theorem 7.14 implies that $\mathsf{CEP}(G)$ belongs to DET. □

Actually, Theorem 7.16 can be slightly extended to groups that are (f.g. nilpotent)-by-(finite solvable) (i.e., groups that have a normal subgroup, which is f.g. nilpotent, and where the quotient is finite solvable). This follows from Theorem 7.14 and the fact that circuit evaluation for a finite solvable group belongs to DET (Theorem 5.15).

7.4 The uniform circuit evaluation problem for unitriangular groups

For Lemma 7.11 it is crucial that the dimension d is a constant. In this section, we consider a uniform variant of the circuit evaluation problem for $\mathsf{UT}_d(\mathbb{Z})$. We denote this problem by $\mathsf{CEP}(\mathsf{UT}_*(\mathbb{Z}))$. The input consists of a unary encoded number d and a circuit, whose inputs are generators of $\mathsf{UT}_d(\mathbb{Z})$ or their inverses. The question is whether the circuit evaluates to the identity matrix. We show that this problem is complete for the complexity class $\mathsf{C}_=\mathsf{LogCFL}$.

Theorem 7.17. *The problem* $\mathsf{CEP}(\mathsf{UT}_*(\mathbb{Z}))$ *is complete for* $\mathsf{C}_=\mathsf{LogCFL}$.

Proof. We start with the upper bound. Consider a circuit $\mathcal{C}$, whose inputs are generators of $\mathsf{UT}_d(\mathbb{Z})$ or their inverses. The dimension d is clearly bounded by the input size. Take the circuit $\mathcal{D}$ over $(\mathbb{Z}, +, \cdot)$ constructed from $\mathcal{C}$ in Lemma 7.12. The formal degree $\mathsf{deg}(\mathcal{D})$ is bounded by $2(d-1)$, i.e., polynomially bounded in the input length. So by Lemma 5.13 the problem to decide whether $[\mathcal{D}] = 0$ is in $\mathsf{C}_=\mathsf{LogCFL}$.

Let us now show that $\mathsf{CEP}(\mathsf{UT}_*(\mathbb{Z}))$ is hard for $\mathsf{C}_=\mathsf{LogCFL}$. Let $(\mathcal{C}_{1,n})_{n\geq 0}$ and $(\mathcal{C}_{2,n})_{n\geq 0}$ be two logspace-uniform families of positive circuits of polynomially bounded size and formal degree over $(\mathbb{N}[x_1, \ldots, x_n], +, \cdot)$. Let $w = a_1 a_2 \cdots a_n \in \{0,1\}^n$ be an input for the circuits $\mathcal{C}_{1,n}$ and $\mathcal{C}_{2,n}$. Let $\mathcal{C}_i$ be the positive circuit obtained from $\mathcal{C}_{i,n}$ by replacing every x_j-labeled input gate by $a_j \in \{0,1\}$. As remarked in Section 5.3 the question whether $[\mathcal{C}_1] = [\mathcal{C}_2]$ is hard for $\mathsf{C}_=\mathsf{LogCFL}$. By [7, Lemma 3.2] we can assume that every gate of $\mathcal{C}_i$ is labeled by its formal degree. By adding (if necessary) additional multiplication gates where one input is set to 1, we can assume that $\mathcal{C}_1$ and $\mathcal{C}_2$ have the same formal degree $d \leq p(n)$ for a polynomial p. Analogously, we can assume that if A is an addition gate in $\mathcal{C}_1$ or $\mathcal{C}_2$ with right-hand side $B + C$, then $\mathsf{deg}(B) = \mathsf{deg}(C) = \mathsf{deg}(A)$. All these preprocessing steps can be carried out in logarithmic space.

We will construct in logarithmic space a circuit $\mathcal{D}$ with inputs from $\Gamma_{d+1} \cup \Gamma_{d+1}^{-1}$ where $\Gamma_{d+1} = \{T_{i,i+1} \mid 1 \leq i < d+1\}$ is our canonical generating set for the matrix group $\mathsf{UT}_{d+1}(\mathbb{Z})$, such that $\mathcal{D}$ evaluates to the identity matrix if and only if $\mathcal{C}_1$ and $\mathcal{C}_2$ evaluate to the same number. Let v_i be the output value of $\mathcal{C}_i$. We first construct in logarithmic space a circuit $\mathcal{D}_1$ that evaluates to the matrix $T_{1,d}^{v_1}$. In the same way we can construct in logarithmic space a second circuit $\mathcal{D}_2$ that evaluates to $T_{1,d}^{-v_2}$. Then, by concatenating the two circuits $\mathcal{D}_1$ and $\mathcal{D}_2$ in a new output gate we obtain the desired circuit.

The gates of $\mathcal{D}_1$ are $A_{i,j}^b$ where A is a gate of $\mathcal{C}_1$, $b \in \{-1, 1\}$, and $1 \leq i < j \leq d$ such that $j - i$ is the formal degree of A. The circuit $\mathcal{D}_1$ will be constructed in such a way that $[A_{i,j}^b]_{\mathcal{D}_1} = T_{i,j}^{b \cdot v}$ where $v = [A]_{\mathcal{C}_1}$. If $\mathsf{rhs}_{\mathcal{C}_1}(A) = 0$, then we set $\mathsf{rhs}_{\mathcal{D}_1}(A_{i,j}^b) = \mathsf{Id}$ and if $\mathsf{rhs}_{\mathcal{C}_1}(A) = 1$, then we set $\mathsf{rhs}_{\mathcal{D}_1}(A_{i,j}^b) = T_{i,j}^b$. Correctness is obvious in these cases. If $\mathsf{rhs}_{\mathcal{C}_1}(A) = B + C$, then we set $\mathsf{rhs}_{\mathcal{D}_1}(A_{i,j}^b) = B_{i,j}^b C_{i,j}^b$. Correctness follows immediately by induction. Note that $\mathsf{deg}(B) = \mathsf{deg}(C) = \mathsf{deg}(A) = j - i$, which ensures that the gates $B_{i,j}^b$ and $C_{i,j}^b$ exist. Finally, if $\mathsf{rhs}_{\mathcal{C}_1}(A) = B \cdot C$, then we set $\mathsf{rhs}_{\mathcal{D}_1}(A_{i,j}^1) = B_{i,k}^{-1} C_{k,j}^{-1} B_{i,k}^1 C_{k,j}^1$ and $\mathsf{rhs}_{\mathcal{D}_1}(A_{i,j}^{-1}) = C_{k,j}^{-1} B_{i,k}^{-1} C_{k,j}^1 B_{i,k}^1$ where k is such that $\mathsf{deg}(B) = k - i$ and $\mathsf{deg}(C) = j - k$. Such a k must exist since $j - i = \mathsf{deg}(A) = \mathsf{deg}(B) + \mathsf{deg}(C)$. Correctness follows from Lemma 3.36 and induction.

□

7.5 Some wreath products with circuit evaluation in coRNC^2 and DET

In this section, we apply the results from the previous sections in this chapter to find groups for which the circuit evaluation problem belongs to coRNC^2 resp. to DET. For wreath products we use the following lemma:

Lemma 7.18. *For every $k \geq 1$ and every finitely generated group G, $\mathsf{CEP}(G \wr \mathbb{Z}^k)$ is NC^2-reducible to $\mathsf{CEP}(G \wr \mathbb{Z})$.*

Proof. The idea is similar to the proof of Lemma 5.26. Let G be generated by the finite set Σ. Fix the generating set $\{a_1, a_2, \ldots, a_k\}$ for $\mathbb{Z}^k$ where every a_i generates a $\mathbb{Z}$-copy. Then $G \wr \mathbb{Z}^k$ is generated by the set $\Gamma = \Sigma \cup \{a_1, a_2, \ldots, a_k\}$. Let $\mathcal{C}$ be a circuit with inputs from $\Gamma \cup \Gamma^{-1}$. First, by an addition circuit we compute in DET the number $d = 2(|\mathsf{val}(\mathcal{G}(\mathcal{C}))| + 1)$. Note that for all $a_i, b_i \in \mathbb{Z}$ $(1 \leq i \leq k)$ with $|a_i|, |b_i| \leq |\mathsf{val}(\mathcal{G}(\mathcal{C}))|$ we have: $(a_1, \ldots, a_k) = (b_1, \ldots, b_k)$ if and only if $\sum_{i=1}^k a_i \cdot d^{i-1} = \sum_{i=1}^k b_i \cdot d^{i-1}$.

From our circuit $\mathcal{C}$ we construct a new circuit $\mathcal{D}$ by replacing every occurrence of a_i (resp., a_i^{-1}) in a right-hand side by a new gate that evaluates to $a^{d^{i-1}}$ (resp., $a^{-d^{i-1}}$). This implies the following: if $(f, (z_1, \ldots, z_k)) = [\mathcal{C}]$ (resp., $(h, z) = [\mathcal{D}]$), then $z = \sum_{i=1}^k z_i \cdot d^{i-1}$ and for all

$(x_1, \ldots, x_k) \in \mathbb{Z}^k$, $f(x_1, \ldots, x_k) = h(x)$ where $x = \sum_{i=1}^{k} x_i \cdot d^{i-1}$. It follows that $[\mathcal{C}] = 1$ in $G \wr \mathbb{Z}^k$ if and only if $[\mathcal{D}] = 1$ in $G \wr \mathbb{Z}$.

□

By Lemma 3.40 and Lemma 7.18 the circuit evaluation problem for a group $(G \times H) \wr \mathbb{Z}^n$ can be reduced in NC^2 to the circuit evaluation problem for the groups $G \wr \mathbb{Z}$ and $H \wr \mathbb{Z}$. Together with Theorem 6.1, Corollary 7.8 and Corollary 7.9 we obtain the following result:

Corollary 7.19. *Let G be a finite direct product of copies of $\mathbb{Z}$ and $\mathbb{Z}_p$ for primes p. Then, for every $n \geq 1$, $\mathsf{CEP}(G \wr \mathbb{Z}^n)$ belongs to coRNC^2.*

Theorem 7.20. *Let G be a finite direct product of copies of $\mathbb{Z}$ and $\mathbb{Z}_p$ for primes p and let H be f.g. virtually abelian. Then $\mathsf{CEP}(G \wr H)$ belongs to coRNC^2. If H is furthermore finite, then $\mathsf{CEP}(G \wr H)$ belongs to DET.*

Proof. Let $K \leq H$ be a f.g. abelian subgroup of finite index m in H. Moreover, let either $K = 1$ (in the case that H is finite) or $K \cong \mathbb{Z}^k$ for some $k \geq 1$. By Lemma 3.41, $G^m \wr K$ is isomorphic to a subgroup of index m in $G \wr H$. Hence by Theorem 7.14, it suffices to show that $\mathsf{CEP}(G^m \wr K)$ belongs to coRNC^2 (resp. DET). In the case $K = \mathbb{Z}^n$ this follows by Corollary 7.19, since G^m is a finite direct product of copies of $\mathbb{Z}$ and $\mathbb{Z}_p$ for primes p. In the case $K = 1$ the group $(G^m \wr 1)$ is isomorphic to G^m, which is finitely generated nilpotent. So by Theorem 7.16 $\mathsf{CEP}(G \wr H)$ belongs to DET. □

Recall that every finitely generated abelian group is a direct product of copies of $\mathbb{Z}$ and $\mathbb{Z}_n$. It is not clear, whether in Corollary 7.20 we can replace G by an arbitrary finitely generated abelian group. On the other hand, if we apply Theorem 5.22 instead of Theorem 6.1 we obtain the following result:

Corollary 7.21. *Let G be f.g. abelian and let H be f.g. virtually abelian. Then $\mathsf{CEP}(G \wr H)$ belongs to coRP. If H is furthermore finite, then $\mathsf{CEP}(G \wr H)$ belongs to DET.*

Recall that for a subgroup H of a group G, $[H, H]$ denotes the *commutator subgroup* of G. It is the subgroup of G generated by all elements $h_1 h_2 h_1^{-1} h_2^{-1}$ with $h_1, h_2 \in H$. It is well known that if N is a normal subgroup of G, then also $[N, N]$ is a normal subgroup of G. Hence, one can consider the quotient group $G/[N, N]$. The following result of Magnus [60] has many applications in combinatorial group theory.

Theorem 7.22 (Magnus embedding theorem)**.** *Let F_k be a free group of rank k and let N be a normal subgroup of F_k. Then $F_k/[N, N] \leq \mathbb{Z}^k \wr F_k/N$.*

Theorem 7.23. *Let F_k be a free group of rank k and let N be a normal subgroup of F_k such that F_k/N is f.g. virtually abelian. Then $\mathsf{CWP}(F_k/[N, N])$ belongs to coRNC^2.*

Proof. By the Magnus embedding theorem, the group $F_k/[N, N]$ embeds into the wreath product $\mathbb{Z}^k \wr (F_k/N)$. For the latter group, the circuit evaluation problem belongs to coRNC^2 by Corollary 7.20. □

7.6 Circuit evaluation for polycyclic groups

In this section we consider the circuit evaluation problem for polycyclic groups. Since every polycyclic group is f.g. linear, circuit evaluation for a polycyclic group can be reduced to polynomial identity testing over $\mathbb{Z}$ or $\mathbb{Z}_n$ and hence it can be solved in coRP. In this section, we show a lower bound: there exists a strongly polycyclic group G (which is also metabelian) such that $\mathsf{PIT}(\mathbb{Z})$ for powerful skew circuits can be reduced to $\mathsf{CEP}(G)$.

Let us start with a specific example of a polycyclic group. Consider the two matrices

$$g_a = \begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix} \text{ and } h = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \tag{7.1}$$

where $a \in \mathbb{R}$, $a \geq 2$. Let $G_a = \langle g_a, h \rangle \leq \mathsf{GL}_2(\mathbb{R})$. Let us remark that, for instance, the group G_2 is not polycyclic, see e.g. [90, p. 56]. On the other hand, we show the following:

Theorem 7.24. *The group $G = G_{1+\sqrt{2}}$ is polycyclic and metabelian.*

Proof. We show that the commutator subgroup of G is isomorphic to $\mathbb{Z} \times \mathbb{Z}$, which implies the theorem. First we calculate the commutator subgroup of G. It is known that the commutator subgroup of a group generated by two elements g_1, g_2 is generated by all commutators $g_1^s g_2^t g_1^{-s} g_2^{-t}$ for $s, t \in \mathbb{Z}$ [67]. Hence,

$$[G, G] = \langle M_{s,t} \mid s, t \in \mathbb{Z} \rangle$$

where for $s, t \in \mathbb{Z}$ we set

$$\begin{aligned}
M_{s,t} &= \begin{pmatrix} 1+\sqrt{2} & 0 \\ 0 & 1 \end{pmatrix}^s \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^t \begin{pmatrix} 1+\sqrt{2} & 0 \\ 0 & 1 \end{pmatrix}^{-s} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-t} \\
&= \begin{pmatrix} (1+\sqrt{2})^s & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} (1+\sqrt{2})^{-s} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -t \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} (1+\sqrt{2})^s & t(1+\sqrt{2})^s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} (1+\sqrt{2})^{-s} & -t(1+\sqrt{2})^{-s} \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & -t + t(1+\sqrt{2})^s \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & t((1+\sqrt{2})^s - 1) \\ 0 & 1 \end{pmatrix}.
\end{aligned}$$

With the setting

$$u = \begin{pmatrix} 1 & \sqrt{2} \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad v = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$$

we show that $\langle M_{s,t} \mid s, t \in \mathbb{Z} \rangle = \langle u, v \rangle$. Moreover, it is easy to see that u and v generate a copy of $\mathbb{Z} \times \mathbb{Z}$.

We have $M_{1,1} = u$ and

$$M_{2,1} M_{1,1}^{-2} = \begin{pmatrix} 1 & 2+2\sqrt{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2\sqrt{2} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = v.$$

This shows that $\langle u, v \rangle \subseteq \langle M_{s,t} \mid s, t \in \mathbb{Z} \rangle$. For the other inclusion assume first that $s \geq 0$. Then

$$\begin{aligned}
t\left(\left(1+\sqrt{2}\right)^s - 1\right) &= t\left(\left(\sum_{i=0}^{s} \binom{s}{i} \sqrt{2}^i\right) - 1\right) \\
&= t\left(\sum_{i=1}^{s} \binom{s}{i} \sqrt{2}^i\right) \\
&= t\left(\sum_{i=1}^{\lfloor \frac{s}{2} \rfloor} \binom{s}{2i} \sqrt{2}^{2i} + \sum_{i=1}^{\lceil \frac{s}{2} \rceil} \binom{s}{2i-1} \sqrt{2}^{2i-1}\right) \\
&= 2\left(\sum_{i=1}^{\lfloor \frac{s}{2} \rfloor} t\binom{s}{2i} 2^{i-1}\right) + \sqrt{2}\left(\sum_{i=1}^{\lceil \frac{s}{2} \rceil} t\binom{s}{2i-1} 2^{i-1}\right).
\end{aligned}$$

So with

$$c_1 = \sum_{i=1}^{\lfloor \frac{s}{2} \rfloor} t\binom{s}{2i} 2^{i-1} \in \mathbb{Z} \quad \text{and} \quad c_2 = \sum_{i=1}^{\lceil \frac{s}{2} \rceil} t\binom{s}{2i-1} 2^{i-1} \in \mathbb{Z}$$

we get

$$M_{s,t} = \begin{pmatrix} 1 & t((1+\sqrt{2})^s - 1) \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2c_1 + \sqrt{2}c_2 \\ 0 & 1 \end{pmatrix} = v^{c_1} u^{c_2}.$$

For $s < 0$ we get with $a = (-s) \bmod 2$:

$$\begin{aligned}
t\left(\left(1+\sqrt{2}\right)^s - 1\right) &= t\left(\left(\sqrt{2}-1\right)^{-s} - 1\right) \\
&= t\left(\left(\sum_{i=0}^{-s}\binom{-s}{i}(\sqrt{2})^i(-1)^{-s-i}\right) - 1\right) \\
&= t\left(-2a + \sum_{i=1}^{-s}\binom{-s}{i}(\sqrt{2})^i(-1)^{-s-i}\right) \\
&= t\left(-2a + \sum_{i=1}^{\lfloor\frac{-s}{2}\rfloor}\binom{-s}{2i}(\sqrt{2})^{2i}(-1)^{-s-2i}\right) + \\
&\quad t\sum_{i=1}^{\lceil\frac{-s}{2}\rceil}\binom{-s}{2i-1}(\sqrt{2})^{2i-1}(-1)^{-s-(2i-1)} \\
&= 2\left(-at + \sum_{i=1}^{\lfloor\frac{-s}{2}\rfloor} t\binom{-s}{2i}2^{i-1}(-1)^{-s-2i}\right) + \\
&\quad \sqrt{2}\left(\sum_{i=1}^{\lceil\frac{-s}{2}\rceil} t\binom{-s}{2i-1}2^{i-1}(-1)^{-s-(2i-1)}\right).
\end{aligned}$$

So with

$$c_1 = -at + \sum_{i=1}^{\lfloor\frac{-s}{2}\rfloor} t\binom{-s}{2i}2^{i-1}(-1)^{-s-2i} \in \mathbb{Z}$$

and

$$c_2 = \sum_{i=1}^{\lceil\frac{-s}{2}\rceil} t\binom{-s}{2i-1}2^{i-1}(-1)^{-s-(2i-1)} \in \mathbb{Z}$$

we get

$$M_{s,t} = \begin{pmatrix} 1 & t((1+\sqrt{2})^s - 1) \\ 0 & 1 \end{pmatrix} = v^{c_1}u^{c_2}.$$

This shows that $\langle M_{s,t} \mid s,t \in \mathbb{Z}\rangle \subseteq \langle u, v\rangle$.

□

The main result of this section is the following:

Theorem 7.25. *Let $a \geq 2$. $\mathsf{PIT}(\mathbb{Z})$ for powerful skew circuits is logspace-reducible to circuit evaluation for the group G_a.*

In particular, there exist polycyclic groups for which the circuit evaluation problem is at least as hard as $\mathsf{PIT}(\mathbb{Z})$ for powerful skew circuits. Recall that it is not known, whether there exists a polynomial time algorithm for $\mathsf{PIT}(\mathbb{Z})$ restricted to powerful skew circuits. Furthermore the theorem shows that circuit evaluation for the Baumslag-Solitar group $\mathsf{BS}_{1,2} := \langle a, t \mid t^{-1}at = a^2\rangle$ is as least as hard as $\mathsf{PIT}(\mathbb{Z})$ for powerful skew circuits, since $\mathsf{BS}_{1,2}$ is exactly the group G_2.

For the proof of Theorem 7.25, we will make use of the following two lemmata. The first one is a result from [6] (see the proof of Proposition 2.2 in [6], where the result is shown for $a = 2$, but the proof immediately generalizes to any $a \geq 2$):

Lemma 7.26. *Let $\mathcal{C}$ be a circuit of size n over $\mathbb{Z}[x_1, \ldots, x_m]$ and $p(x_1, \ldots, x_m) = [\mathcal{C}]$. Let $a \geq 2$ be a real number. Then $p(x_1, \ldots, x_m)$ is the zero-polynomial if and only if $p(\alpha_1, \ldots, \alpha_m) = 0$ where $\alpha_i = a^{2^{i \cdot n^2}}$ for $1 \leq i \leq m$.*

In the second lemma we eliminate every multiplication gate with a negative input gate:

Lemma 7.27. *Let $\mathcal{C} = (V, S, \mathsf{rhs})$ be a powerful skew circuit over $(\mathbb{Z}[x], +, \cdot)$. We can transform $\mathcal{C}$ in logarithmic space into an equivalent powerful skew circuit $\mathcal{D} = (V', S, \mathsf{rhs}')$ such that every multiplication gate $A \in V'$ is of the form $\mathsf{rhs}'(A) = x^n \cdot B$ for some gate $B \in V'$ and $n \in \mathbb{N}$.*

Proof. As for skew circuits we can assume that every input gate is either an input of addition gates or an input of multiplication gates. With a similar construction as in the proof of Lemma 5.19 we introduce for every gate $A \in V$ a gate A' such that $[A'] = -[A]$. In detail we set $V' = V \cup \{A' \mid A \in V\}$ and define rhs' in the following way:

$$\mathsf{rhs}'(A) = \begin{cases} x^n \cdot B' & \text{if } \mathsf{rhs}(A) = -x^n \cdot B \text{ for } B \in V \\ \mathsf{rhs}(A) & \text{in every other case} \end{cases}$$

and

$$\mathsf{rhs}'(A') = \begin{cases} B' + C' & \text{if } \mathsf{rhs}(A) = B + C \\ -\mathsf{rhs}(A) & \text{if } A \text{ is an input-gate of addition gates} \\ x^n \cdot B' & \text{if } \mathsf{rhs}(A) = x^n \cdot B \\ x^n \cdot B & \text{if } \mathsf{rhs}(A) = -x^n \cdot B \text{ for } B \in V \end{cases}$$

By induction one gets for every $A \in V$ that $[A] = -[A']$ and so $[\mathcal{C}] = [\mathcal{D}]$. By the definition of rhs' every multiplication gate is of the form $x^n \cdot B$ for some $n \in \mathbb{N}$ and $B \in V'$. □

Proof of Theorem 7.25. Let us fix a powerful skew circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over $\mathbb{Z}[x], +, \cdot)$ of size n. By Lemma 7.27 we can assume that every multiplication gate is of the form $x^m \cdot A$ for some $A \in V$. By Lemma 7.26 we know that $[\mathcal{C}] = p(x) = 0$ if and only if $p(\alpha) = 0$ for $\alpha = a^{2^{n^2}}$. We will define a circuit $\mathcal{D}$ with inputs from $\{g_a, g_a^{-1}, h, h^{-1}\}$ such that $[\mathcal{D}] = \mathsf{Id}$ in G_a if and only if $[\mathcal{C}] = 0$. First of all, using iterated squaring, we can construct for every input of the form x^m a circuit $\mathcal{D}'$ with gates $A_1, A_1^{-1}, \ldots, A_m, A_m^{-1}$ (and some other auxiliary gates) such that

$$\begin{aligned} [A_i]_{\mathcal{D}'} &= g_a^{i \cdot 2^{n^2}} = \begin{pmatrix} a^{i \cdot 2^{n^2}} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \alpha^i & 0 \\ 0 & 1 \end{pmatrix} \text{ and} \\ [A_i^{-1}]_{\mathcal{D}'} &= g_a^{-i \cdot 2^{n^2}} = \begin{pmatrix} a^{-i \cdot 2^{n^2}} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \alpha^{-i} & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

We now construct the circuit $\mathcal{D}$ as follows: the set of gates of $\mathcal{D}$ consists of the gates of $\mathcal{C}$ and the gates of $\mathcal{D}'$. We copy the right-hand sides from $\mathcal{D}'$ and define the right-hand side for a gate A of $\mathcal{C}$ as follows:

$$\mathsf{rhs}_{\mathcal{D}}(A) = \begin{cases} h & \text{if } \mathsf{rhs}_{\mathcal{C}}(A) = 1 \\ h^{-1} & \text{if } \mathsf{rhs}_{\mathcal{C}}(A) = -1 \\ A_m h^b A_m^{-1} & \text{if } \mathsf{rhs}_{\mathcal{C}}(A) = bx^m \text{ for } b \in \{-1, 1\} \\ BC & \text{if } \mathsf{rhs}_{\mathcal{C}}(A) = B + C \\ A_m B A_m^{-1} & \text{if } \mathsf{rhs}_{\mathcal{C}}(A) = x^m \cdot B \end{cases}$$

We claim that for every gate A of $\mathcal{C}$ we have the following, where we denote for better readability the polynomial $[A]_{\mathcal{C}}$ to which gate A evaluates by p_A:

$$[A]_{\mathcal{D}} = \begin{pmatrix} 1 & p_A(\alpha) \\ 0 & 1 \end{pmatrix}$$

The case $\mathsf{rhs}_{\mathcal{C}}(A) \in \{-1, 1\}$ is obvious.

If $\mathsf{rhs}_{\mathcal{C}}(A) = x^m$, then we obtain

$$\begin{aligned}
[A]_{\mathcal{D}} &= \begin{pmatrix} \alpha^m & 0 \\ 0 & 1 \end{pmatrix} h \begin{pmatrix} \alpha^{-m} & 0 \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \alpha^m & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha^{-m} & 0 \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \alpha^m & \alpha^m \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha^{-m} & 0 \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & \alpha^m \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & p_A(\alpha) \\ 0 & 1 \end{pmatrix}.
\end{aligned}$$

If $\mathsf{rhs}_{\mathcal{C}}(A) = B + C$, then we obtain by induction

$$\begin{aligned}
[A]_{\mathcal{D}} &= [B]_{\mathcal{D}}[C]_{\mathcal{D}} \\
&= \begin{pmatrix} 1 & p_B(\alpha_1, \ldots, \alpha_m) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & p_C(\alpha_1, \ldots, \alpha_m) \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & p_B(\alpha_1, \ldots, \alpha_m) + p_C(\alpha_1, \ldots, \alpha_m) \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & p_A(\alpha_1, \ldots, \alpha_m) \\ 0 & 1 \end{pmatrix}.
\end{aligned}$$

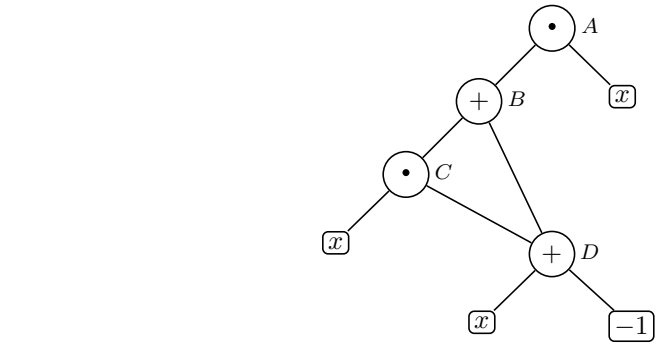
Finally, if $\mathsf{rhs}_{\mathcal{C}}(A) = x^m \cdot B$ then we obtain by induction

$$\begin{aligned}
[A]_{\mathcal{D}} &= \begin{pmatrix} \alpha^m & 0 \\ 0 & 1 \end{pmatrix} [B]_{\mathcal{D}} \begin{pmatrix} \alpha^{-m} & 0 \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \alpha^m & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & p_B(\alpha) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha^{-m} & 0 \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \alpha^m & \alpha^m \cdot p_B(\alpha_1, \ldots, \alpha_m) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha^{-m} & 0 \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & \alpha^m \cdot p_B(\alpha_1, \ldots, \alpha_m) \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & p_A(\alpha) \\ 0 & 1 \end{pmatrix}.
\end{aligned}$$

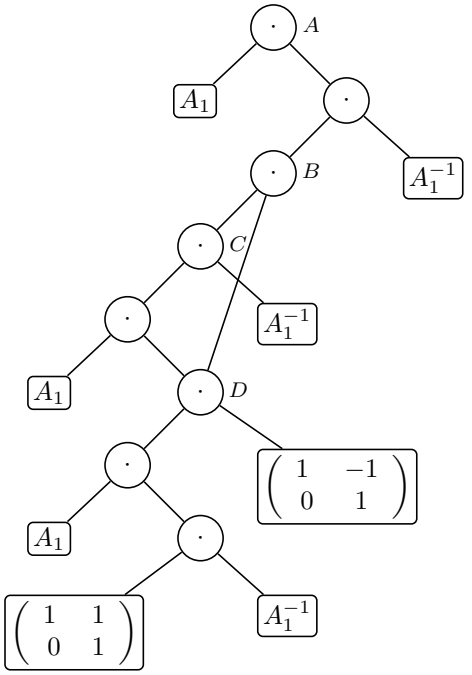
We finally take the output gate S of the circuit $\mathcal{C}$ as the output gate of $\mathcal{D}$. Then, $[\mathcal{D}]$ yields the identity matrix in the group G_a if and only if $p_S(\alpha) = 0$. □

Example 7.28. *In Figure 7.5 the proof of Theorem 7.25 is illustrated. For the sake of convenience we consider a skew circuit (instead of a powerful skew one) $\mathcal{C}$ with $|\mathcal{C}| = 8$. This circuit is transformed into a circuit $\mathcal{D}$ over $G_{1+\sqrt{2}}$. Here $\alpha = (1+\sqrt{2})^{2^{64}}$, so we need 64 multiplication gates plus one input gate to generate the gate A_1 (resp. A_1^{-1}) with $[A_1] = g_{1+\sqrt{2}}^{2^{64}}$ (resp. $[A_1^{-1}] = g_{1+\sqrt{2}}^{-2^{64}}$). For this reason we split the circuit in two parts: on the left there is the circuit $\mathcal{D}$, where A_1 and A_1^{-1} are left as inputs and on the right there are the two circuits that generate these gates.*

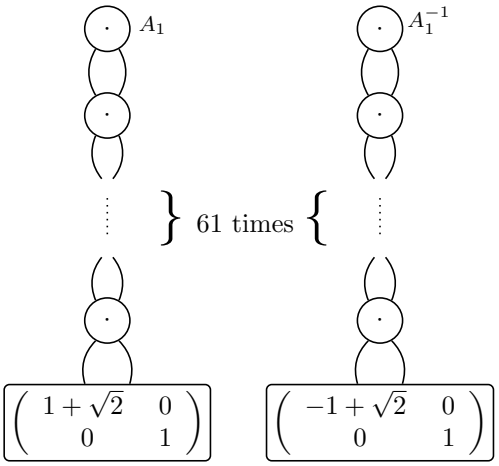
Let us look again at the group $G = G_{1+\sqrt{2}}$ from Theorem 7.24. Its commutator subgroup is isomorphic to $\mathbb{Z} \times \mathbb{Z}$. Moreover, the quotient $G/[G,G]$ is isomorphic to $\mathbb{Z} \times \mathbb{Z}_2$: the G-generator h from (7.1) satisfies $h^2 \in [G,G]$, whereas the generator $g_{1+\sqrt{2}}$ has infinite order in the quotient. Hence, G has a subnormal series of the form $G \rhd H \rhd \mathbb{Z} \times \mathbb{Z} \rhd \mathbb{Z} \rhd 1$, where H has index two in G and $H/(\mathbb{Z} \times \mathbb{Z}) \cong \mathbb{Z}$. The group H is strongly polycyclic and has Hirsch length three. By Theorem 7.14 we obtain the following result:



(a) Skew Circuit $\mathcal{C}$



(b) Circuit over $G_{1+\sqrt{2}}$



(c) Circuits to generate A_1 and A_1^{-1}

Figure 7.5: Transformation of a skew circuit into a circuit over $G_{1+\sqrt{2}}$.

Corollary 7.29. *There is a strongly polycyclic group H of Hirsch length 3 such that* $\mathsf{PIT}(\mathbb{Z})$ *for powerful skew circuits is logspace-reducible to* $\mathsf{CEP}(H)$.

Chapter 8

Circuit evaluation for finite semirings

8.1 Introduction

Recall that for finite semigroups, the following result was shown in [19]:

Let S be a finite semigroup.

- If S is aperiodic, then $\mathsf{CEP}(S)$ is in NL.
- If S is solvable, then $\mathsf{CEP}(S)$ belongs to DET.
- If S is not solvable, then $\mathsf{CEP}(S)$ is P-complete.

One way to extend this result was shown in [70], where the authors considered circuit evaluation for finite groupoids, i.e., the operation $\cdot$ does not need to be associative. They showed that for so-called polyabelian groupoids circuit evaluation is in DET, but there are also non-polyabelian groupoids with a circuit evaluation problem in DET. It is still an open problem, whether one can reach a similar dichotomy result as for semigroups for this setting. In this section we extend the semigroup result to finite semirings. This means we split the class of semirings in two types: one type where circuit evaluation is P-complete and one where circuit evaluation is in DET. The results of this chapter have appeared in [3]. Note that $\mathsf{CEP}(\mathcal{R}^+)$ (resp., $\mathsf{CEP}(\mathcal{R}^\bullet)$) is the restriction of $\mathsf{CEP}(R)$ to circuits without multiplication (resp., addition) gates. Since every commutative semigroup is solvable, the result above implies that for a finite semiring R circuit evaluation for $\mathcal{R}^+$ belongs to DET. Obviously, if $\mathcal{R}^\bullet$ is not solvable, then $\mathsf{CEP}(R)$ is P-complete, but there are also semirings with a solvable multiplicative semigroup with P-complete circuit evaluation problem as the following two examples show:

one example is Ladner's classical P-completeness result for the Boolean circuit value problem stated in Theorem 5.14.

The other example is circuit evaluation for the finite semiring $(\mathbb{Z}_n, +, \cdot)$:

Lemma 8.1. *Let $n \geq 2$ and $R = (\mathbb{Z}_n, +, \cdot)$. Then $\mathsf{CEP}(R)$ is P-complete.*

Proof. We can reduce the Boolean circuit value problem over $\{0, 1, \wedge, \neg\}$ (which is also known to be P-complete) to $\mathsf{CEP}(R)$: a gate $z = x \wedge y$ is replaced by $z = x \cdot y$ and a gate $y = \neg x$ is replaced by $y = 1 + (n-1) \cdot x$. □

In this section we show that these are in fact the only essential examples for semirings with a solvable multiplicative semigroup and a P-complete circuit evaluation problem and that circuits over every finite semiring with a solvable multiplicative semigroup that does not contain $\mathbb{B}_2$ or $\mathbb{Z}_n$ as a subsemiring can be evaluated in DET.

By Lemma 3.25 we know that the last property is equivalent to the case R is $\{0,1\}$-free. This leads to the following main result of this section:

Theorem 8.2. *If the finite semiring R is $\{0,1\}$-free, then circuit evaluation for R belongs to $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(\mathcal{R}^+), \mathsf{CEP}(\mathcal{R}^\bullet))$. Otherwise $\mathsf{CEP}(R)$ is P-complete.*

By the previously mentioned lemmata the P-completeness part is clear. In the rest of this section we prove the reduction part.

Theorem 5.15 and Theorem 8.2 yield the following corollary:

Corollary 8.3. *Let R be a finite semiring.*

- *If R is $\{0,1\}$-free and $\mathcal{R}^\bullet$ is solvable, then $\mathsf{CEP}(R)$ belongs to DET.*
- *If R is not $\{0,1\}$-free or $\mathcal{R}^\bullet$ is not solvable, then $\mathsf{CEP}(R)$ is P-complete.*

8.2 Circuits over $\{0,1\}$-free semirings

The proof of the reduction part of Theorem 8.2 will proceed in two steps. In the first step we reduce the problem to the evaluation of circuits in which the computation admits a type-function defined in the following. In the second step, we show how to evaluate such circuits. Throughout this section we fix a finite semiring $(R,+,\cdot)$ of size $n = |R|$.

Definition 8.4 (type-function)**.** *Let $E = E(R)$ be the set of multiplicative idempotents. Let $\mathcal{C} = (V, \mathsf{rhs})$ be a circuit over R such that $[A]_\mathcal{C} \in ERE$ for all $A \in V$. A type-function for $\mathcal{C}$ is a mapping $\mathsf{type} : V \to E \times E$ such that for all gates $A \in V$:*

- *If $\mathsf{type}(A) = (e,f)$, then $[A]_\mathcal{C} \in eRf$.*
- *If A is an addition gate with $\mathsf{rhs}(A) = B + C$, then $\mathsf{type}(A) = \mathsf{type}(B) = \mathsf{type}(C)$.*
- *If A is a multiplication gate with $\mathsf{rhs}(A) = B \cdot C$, $\mathsf{type}(B) = (e,e')$, and $\mathsf{type}(C) = (f',f)$, then $\mathsf{type}(A) = (e,f)$.*

A circuit is called type admitting if it admits a type-function.

Note that we do not need an output gate in a type admitting circuit.

Definition 8.5 (affine function)**.** *A function $\alpha : R^m \to R$ ($m \geq 0$) is called affine if there are $a_1, b_1, \ldots, a_m, b_m, c \in R$ such that for all $x_1, \ldots, x_m \in R$:*

$$\alpha(x_1, \ldots, x_m) = \sum_{i=1}^{m} a_i x_i b_i + c \quad \text{or} \quad \alpha(x_1, \ldots, x_m) = \sum_{i=1}^{m} a_i x_i b_i.$$

We represent this affine function by the tuple $(a_1, b_1, \ldots, a_m, b_m, c)$ or $(a_1, b_1, \ldots, a_m, b_m)$. The two main lemmata we prove in Section 8.2.1 and in Section 8.2.2 are the following:

Lemma 8.6. *Given a circuit $\mathcal{C}$ over the finite semiring R, one can compute in $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(\mathcal{R}^+))$*

- *an affine function $\alpha : R^m \to R$ for some $0 \leq m \leq |R|^4$,*
- *a type admitting circuit $\mathcal{C}' = (V', \mathsf{rhs}')$, and*
- *a list of gates $A_1, \ldots, A_m \in V'$ such that $[\mathcal{C}] = \alpha([A_1]_{\mathcal{C}'}, \ldots, [A_m]_{\mathcal{C}'})$.*

Lemma 8.7. *If R is $\{0,1\}$-free, then the restriction of $\mathsf{CEP}(R)$ to type admitting circuits is in $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(\mathcal{R}^+), \mathsf{CEP}(\mathcal{R}^\bullet))$.*

Theorem 8.2 is an immediate corollary of the two lemmata above and the obvious fact that an affine function with a constant number of inputs can be evaluated in AC^0.

It is not clear how to test efficiently whether a circuit is type admitting. But this is not a problem for us, since we will apply Lemma 8.7 only to circuits resulting from Lemma 8.6, which are type admitting by construction. Notice that in the case $\mathcal{R}^\bullet$ is a monoid every circuit over R is already type admitting for the type-function $\mathsf{type}(A) = (1,1)$ for every gate A. So Lemma 8.6 is only needed in the case $\mathcal{R}^\bullet$ is a semigroup without an identity element.

8.2.1 Step 1: reduction to type admitting circuits

In this section, we prove Lemma 8.6. Let $\mathcal{C}$ be a circuit over our fixed finite semiring $R = (R, +, \cdot)$ of size $n = |R|$. We assume that $n \geq 2$ (the case $n = 1$ is trivial). Throughout this section we will use $E = E(R)$. Note that $R^n = RER$ is closed under multiplication with elements from R. Thus, $\langle R^n \rangle$ is an ideal ($\langle R^n \rangle$ denotes the subsemiring that is additively generated by elements from R^n). Every element $a \in \langle R^n \rangle$ can be written as a finite sum $a = \sum_{i=1}^{k} a_i$ with $a_i \in R^n$. Moreover, since R is a fixed finite semiring, the number k of summands can be bounded by a constant that only depends on R.

The reduction to type admitting circuits is done in two steps:

$$\text{circuit over } R \xrightarrow{\text{Lemma 8.10}} \text{circuit over } \langle R^n \rangle = \langle RER \rangle \xrightarrow{\text{Lemma 8.11}} \text{type admitting circuit}$$

Before we state and prove Lemma 8.10 and Lemma 8.11, we show a lemma that allows to eliminate certain input values from a circuit:

Lemma 8.8. *Assume that $I \subseteq R$ is a non-empty ideal of R. Let $\mathcal{C} = (V, S, \mathsf{rhs})$ be a circuit over R. Consider the set $U = \{A \in V \mid A \text{ is an inner gate or } \mathsf{rhs}(A) \in I\}$ and assume that $S \in U$ and for all $A, B, C \in V$ the following holds:*

- *If $\mathsf{rhs}(A) = B \cdot C$ then $B \in U$ or $C \in U$.*
- *If $\mathsf{rhs}(A) = B + C$ then $B, C \in U$.*

Then there is a logspace-computable function that returns for a given circuit $\mathcal{C}$ with the above properties an equivalent circuit $\mathcal{D}$ over the ideal (and hence subsemiring) I.

Proof. Let $U \subseteq V$ be defined as in the lemma. We first compute in logarithmic space a circuit $\mathcal{C}' = (V', (S)_{1,1}, \mathsf{rhs}')$ which contains a gate $A_{\ell,r}$ for every gate $A \in U$ and $\ell, r \in R^1$ (recall that R^1 is R together with a fresh multiplicative identity 1) such that $[A_{\ell,r}]_{\mathcal{C}'} = \ell \cdot [A]_{\mathcal{C}} \cdot r$. Note that V' is non-empty since $S \in U$. The gates of $\mathcal{C}'$ are indexed by elements of R^1 instead of R to simplify the notation in the following. To define the right-hand sides let us take a gate $A \in U$ and $\ell, r \in R^1$.

Case 1. $\mathsf{rhs}(A) = a \in I$. We set $\mathsf{rhs}'(A_{\ell,r}) = \ell a r$. Note that $\ell a r \in I$, since I is an ideal in R.

Case 2. $\mathsf{rhs}(A) = B + C$. Then we must have $B, C \in U$ and we set $\mathsf{rhs}'(A_{\ell,r}) = B_{\ell,r} + C_{\ell,r}$.

Case 3. $\mathsf{rhs}(A) = B \cdot C$. Then $B \in U$ or $C \in U$. If both B and C belong to U, then we set $\mathsf{rhs}'(A_{\ell,r}) = B_{\ell,1} \cdot C_{1,r}$. If $C \notin U$ then $B \in U$ and $\mathsf{rhs}(C) = c \in R \setminus I$. We set $\mathsf{rhs}'(A_{\ell,r}) = B_{\ell,cr}$. The case $B \notin U$ is symmetric.

The correctness of the above construction can be easily shown by induction along the partial order $\leq_{\mathcal{C}}$. □

Figure 8.1 illustrates the construction for $R = \{a, b, c\}$ and the ideal $I = \{c\}$ (the concrete semiring structure is not important). Note that $a^2cb \in I$. The circuit on the right-hand side is only the part of the constructed circuit that is connected to the gate $A_{1,1}$. Copy gates are left unlabeled.

For the following proofs, it is sometimes more convenient to consider a circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over the free semiring $\mathbb{N}[R]$ generated by the set R.[1] Recall that this semiring consists of all mappings $f : R^+ \to \mathbb{N}$ with finite and non-empty support where R^+ consists of all finite non-empty words over the alphabet R. So, there are two ways to evaluate $\mathcal{C}$: we can evaluate $\mathcal{C}$ over R (and this is our main interest) and we can evaluate $\mathcal{C}$ over $\mathbb{N}[R]$. In order to distinguish these two ways of evaluation, we write $[\![A]\!]_{\mathcal{C}} \in \mathbb{N}[R]$ for the value of gate $A \in V$ in $\mathcal{C}$, when $\mathcal{C}$ is evaluated in $\mathbb{N}[R]$. Again we omit the index $\mathcal{C}$ if it is clear from the context. Moreover, $[\![\mathcal{C}]\!] = [\![S]\!]_{\mathcal{C}}$. Note

[1] Of course, there is no chance of efficiently evaluating a circuit over the free semiring $\mathbb{N}[R]$, since this might produce a doubly exponential number of monomials of exponential length. Circuit evaluation over $\mathbb{N}[R]$ is only used as a tool in our proofs.

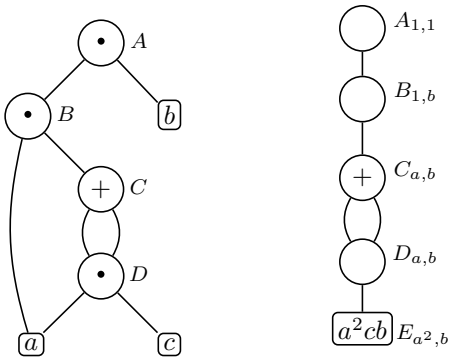


Figure 8.1: Illustration of the proof of Lemma 8.8 with the ideal $I = \{c\}$ in $R = \{a, b, c\}$.

that $[\![A]\!]_{\mathcal{C}}$ is a mapping from R^+ to $\mathbb{N}$; hence for a word $w \in R^+$, $[\![A]\!]_{\mathcal{C}}(w)$ is a natural number. Let h be the canonical semiring homomorphism from $\mathbb{N}[R]$ to R that evaluates a non-commutative polynomial in the semiring R. Thus, we have $[A]_{\mathcal{C}} = h([\![A]\!]_{\mathcal{C}})$ for every gate A and $[\mathcal{C}] = h([\![\mathcal{C}]\!])$. An example of a free evaluation of a circuit is shown in Figure 8.2 on the left.

Recall that $|R| = n \geq 2$. We define

$$R^{<n} = \{w \in R^+ \mid |w| < n\} \quad \text{and} \quad R^{\geq n} = \{w \in R^+ \mid |w| \geq n\}.$$

Note that these are sets of finite words over the alphabet R. So, these notations should not be confused with the notation R^n, which is a subset of R (the set of all n-fold products). In fact, we have $h(R^{\geq n}) = R^n = RER$. For every non-commutative polynomial $f \in \mathbb{N}[R]$ we define $f^\sigma, f^\lambda \in \mathbb{N}[R] \cup \{\bot\}$ (the short part and the long part of f where $\bot$ is a new symbol that stands for "undefined") as follows:

1. If $\mathsf{supp}(f) \subseteq R^{<n}$, then $f^\sigma = f$ and $f^\lambda = \bot$.
2. If $\mathsf{supp}(f) \subseteq R^{\geq n}$, then $f^\sigma = \bot$ and $f^\lambda = f$.
3. Otherwise let $f^\sigma, f^\lambda \in \mathbb{N}[R]$ such that $f = f^\sigma + f^\lambda$, $\mathsf{supp}(f^\sigma) \subseteq R^{<n}$ and $\mathsf{supp}(f^\lambda) \subseteq R^{\geq n}$.

Note that $f^\sigma \neq \bot$ or $f^\lambda \neq \bot$, and that the decomposition in 3 is unique. Moreover, if $f^\sigma \neq \bot \neq f^\lambda$, then $f = f^\sigma + f^\lambda$.

Example 8.9. *Let $R = \{a, b, c\}$ and thus $n = 3$. Let $f = 2abbca + 3caab + bab + 4ac + 7b \in \mathbb{N}[\{a, b, c\}]$. We have $f^\sigma = 4ac + 7b$ and $f^\lambda = 2abbca + 3caab + bab$.*

Lemma 8.10. *There is a function in $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(\mathcal{R}^+))$ that returns for a given circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over R either*

- *the semiring element $[\mathcal{C}] \in R$ (namely if $[\![\mathcal{C}]\!]^\lambda = \bot$), or*
- *a circuit $\mathcal{D}$ over the subsemiring $\langle R^n \rangle = \langle RER \rangle$ such that $[\mathcal{C}] = [\mathcal{D}]$ (namely if $[\![\mathcal{C}]\!]^\sigma = \bot$), or*
- *a circuit $\mathcal{D}$ over the subsemiring $\langle R^n \rangle = \langle RER \rangle$ and a semiring element $\sigma \in R$ such that $[\mathcal{C}] = [\mathcal{D}] + \sigma$ (namely if $[\![\mathcal{C}]\!]^\sigma \neq \bot \neq [\![\mathcal{C}]\!]^\lambda$).*

Proof. In the following, we omit the index $\mathcal{C}$ in $[A]_{\mathcal{C}}$ and $[\![A]\!]_{\mathcal{C}}$ where $A \in V$ is a gate of the circuit $\mathcal{C}$.

Step 1. We first compute in $\mathsf{AC}^0(\mathsf{NL})$ the set of all gates $A \in V$ such that $[\![A]\!]^\sigma \neq \bot$. For this, we construct in logarithmic space a circuit $\mathcal{A}$ over the semiring $(\mathbb{N}, +, \cdot)$ with gates A_w where $A \in V$ and $w \in R^{<n}$ such that $[A_w]_{\mathcal{A}} = [\![A]\!](w)$ as follows:

- If $\mathsf{rhs}(A) = a \in R$, then $\mathsf{rhs}(A_w) = \begin{cases} 1 & \text{if } w = a \\ 0 & \text{otherwise} \end{cases}$.
- If $\mathsf{rhs}(A) = B + C$, then $\mathsf{rhs}(A_w) = B_w + C_w$.
- If $\mathsf{rhs}(A) = B \cdot C$, then $\mathsf{rhs}(A_w) = \sum_{w=uv} B_u \cdot C_v$ where the sum goes over all $u, v \in R^{<n}$ with $w = uv$.

Note that the empty sum is interpreted as 0 and that $\mathcal{A}$ has constant multiplication depth. Moreover, the multiplication depth of a gate A_w is $|w|-1$. By Lemma 5.8 with the structure-preserving partition $V_i = \{A_w \mid |w| = i\}$ we can transform in logarithmic space $\mathcal{A}$ into an equivalent addition circuit, which we still denote by $\mathcal{A}$. The circuit $\mathcal{A}$ contains all gates A_w ($A \in V$, $w \in R^{<n}$) and possibly some additional gates.

We can assume that $\mathcal{A}$ has a unique input gate Z with right-hand side 1. Let U_σ be the set of all gates X of $\mathcal{A}$ such that $Z \leq_{\mathcal{A}} X$. These are exactly those gates of $\mathcal{A}$ that evaluate to a number larger than zero. Hence, for all $A \in V$ and $w \in R^{<n}$, we have $A_w \in U_\sigma$ if and only if $[\![A]\!](w) > 0$. Moreover, $[\![A]\!]^\sigma \neq \bot$ if and only if $A_w \in U_\sigma$ for at least one $w \in R^{<n}$. The set U_σ can be computed in $\mathsf{AC}^0(\mathsf{NL})$. Hence, we can also compute for every $A \in V$ the information whether $[\![A]\!]^\sigma \neq \bot$ and, in case $[\![A]\!]^\sigma \neq \bot$, the set $\mathsf{supp}([\![A]\!]^\sigma) = \mathsf{supp}([\![A]\!]) \cap R^{<n}$.

Step 2. For each gate $A \in V$ with $[\![A]\!]^\sigma \neq \bot$ we now compute the semiring element $h([\![A]\!]^\sigma) \in R$. For this we construct in logarithmic space a circuit over $\mathcal{R}^+$ that evaluates to $h([\![A]\!]^\sigma)$. Hence, $h([\![A]\!]^\sigma)$ can be computed using oracle access to $\mathsf{CEP}(\mathcal{R}^+)$.

We first remove from the addition circuit $\mathcal{A}$ all gates that are not in U_σ. Moreover, gate Z is now the only input gate of $\mathcal{A}$. For a semiring element $a \in R$ we define the circuit $\mathcal{C}_a$ (over $\mathcal{R}^+$) by taking the addition circuit $\mathcal{A}$ and redefining $\mathsf{rhs}_\sigma(Z) = a$. Then, for every gate $A_w \in U_\sigma$ ($A \in V$, $w \in R^{<n}$) we have $[A_w]_{\mathcal{C}_a} = [\![A]\!](w) \cdot a$. In particular, if $[\![A]\!]^\sigma \neq \bot$, then

$$h([\![A]\!]^\sigma) = \sum_{w \in \mathsf{supp}([\![A]\!]) \cap R^{<n}} [\![A]\!](w) \cdot h(w) \;=\; \sum_{w \in \mathsf{supp}([\![A]\!]) \cap R^{<n}} [A_w]_{\mathcal{C}_{h(w)}}.$$

From the circuits $\mathcal{C}_{h(w)}$ we can construct in logarithmic space a circuit over $\mathcal{R}^+$ for this semiring element. Evaluating this circuit using oracle access to $\mathsf{CEP}(\mathcal{R}^+)$ yields the element $h([\![A]\!]^\sigma)$.

Step 3. Next, we compute in $\mathsf{AC}^0(\mathsf{NL})$ the set of all gates $A \in V$ such that $[\![A]\!]^\lambda \neq \bot$. Since $n \geq 2$ (our initial assumption on the semiring R), we have $[\![A]\!]^\lambda \neq \bot$ if and only if there exist a gate $A' \leq_{\mathcal{C}} A$ with $\mathsf{rhs}(A') = B \cdot C$ and words $w_1, w_2 \in R^{<n}$ such that $|w_1 w_2| \geq n$, $[\![B]\!](w_1) > 0$ (i.e., $B_{w_1} \in U_\sigma$) and $[\![C]\!](w_2) > 0$ (i.e., $C_{w_2} \in U_\sigma$). This condition can be tested in NL. Hence, we can assume that the set of all $A \in V$ with $[\![A]\!]^\lambda \neq \bot$ is computed. If $[\![S]\!]^\lambda = \bot$, then we must have $[\![S]\!]^\sigma \neq \bot$ and we return the previously computed semiring element $h([\![S]\!]^\sigma)$, which is $[\mathcal{C}]$ in this case. Let us now assume that $[\![S]\!]^\lambda \neq \bot$.

Step 4. We then construct a circuit $\mathcal{C}_\lambda = (V_\lambda, (S)_\lambda, \mathsf{rhs}_\lambda)$, which contains for every gate $A \in V$ with $[\![A]\!]^\lambda \neq \bot$ a gate A_λ such that $[A_\lambda]_{\mathcal{C}_\lambda} = h([\![A]\!]^\lambda)$. In particular, $[\mathcal{C}_\lambda] = h([\![\mathcal{C}]\!]^\lambda)$.

In a first step, we compute in AC^0 the set M^λ that consists of of all multiplication gates $A \in V$ such that the following conditions hold: $[\![A]\!]^\lambda \neq \bot$, $\mathsf{rhs}(A) = B \cdot C$ for $B, C \in V$, $[\![B]\!]^\sigma \neq \bot \neq [\![C]\!]^\sigma$, and there exist $u \in \mathsf{supp}([\![B]\!]^\sigma)$, $v \in \mathsf{supp}([\![C]\!]^\sigma)$ with $|uv| \geq n$. This means that in the product $[\![B]\!] \cdot [\![C]\!]$ a monomial of length at least n arises from monomials $u \in \mathsf{supp}([\![B]\!])$, $v \in \mathsf{supp}([\![C]\!])$, both of which have length smaller than n.

Next, for every multiplication gate $A \in M^\lambda$ where $\mathsf{rhs}(A) = B \cdot C$ we compute in $\mathsf{AC}^0(\mathsf{CEP}(\mathcal{R}^+))$ the semiring element

$$m_A := \sum_{u,v} ([\![B]\!](u)[\![C]\!](v)) \cdot h(uv) \in \langle R^n \rangle$$

where the sum is taken over all words $u \in \mathsf{supp}([\![B]\!]^\sigma)$, $v \in \mathsf{supp}([\![C]\!]^\sigma)$ with $|uv| \geq n$. Let us take the addition circuit $\mathcal{A}$ constructed in Step 1 and Step 2 above. We add to $\mathcal{A}$ a single layer

of multiplication gates $A_{u,v}$ where $B_u, C_v \in U_\sigma$. The right-hand side of $A_{u,v}$ is $B_u \cdot C_v$. Using Lemma 5.8, we can transform this circuit in logarithmic space into an equivalent addition circuit; let us denote this circuit by $\mathcal{A}^2$. Clearly, gate $A_{u,v}$ evaluates to the number $[\![B]\!](u)[\![C]\!](v) \in \mathbb{N}$. This number is larger than zero, since $B_u, C_v \in U_\sigma$. Hence, by replacing in the addition circuit $\mathcal{A}^2$ the input value 1 by the semigroup element $h(uv)$, we obtain a circuit over the semigroup $\mathcal{R}^+$, which we can evaluate using oracle access to $\mathsf{CEP}(\mathcal{R}^+)$. The value of gate $A_{u,v}$ yields the semiring element $([\![B]\!](u)[\![C]\!](v)) \cdot h(uv)$. Finally, the sum of all these values (for all $u \in \mathsf{supp}([\![B]\!]^\sigma)$, $v \in \mathsf{supp}([\![C]\!]^\sigma)$ with $|uv| \geq n$) can be computed by another AC^0-computation (it is a sum of a constant number of semiring elements).

It remains to define the right-hand sides of the gates A_λ in $\mathcal{C}_\lambda$. We distinguish the following cases (note that A must be an inner gate of $\mathcal{C}$ since $n \geq 2$).

Case 1: $\mathsf{rhs}(A) = B + C$. Then we must have $[\![B]\!]^\lambda \neq \bot$ or $[\![C]\!]^\lambda \neq \bot$ (otherwise $[\![A]\!]^\lambda = \bot$) and we set

$$\mathsf{rhs}_\lambda(A_\lambda) = \begin{cases} B_\lambda, & \text{if } [\![C]\!]^\lambda = \bot, \\ C_\lambda, & \text{if } [\![B]\!]^\lambda = \bot, \\ B_\lambda + C_\lambda, & \text{otherwise.} \end{cases}$$

Case 2: $\mathsf{rhs}(A) = B \cdot C$, $A \in M^\lambda$, and $\bot \notin \{[\![B]\!]^\sigma, [\![B]\!]^\lambda, [\![C]\!]^\sigma, [\![C]\!]^\lambda\}$. Then we set

$$\mathsf{rhs}_\lambda(A_\lambda) = B_\lambda \cdot C_\lambda + h([\![B]\!]^\sigma) \cdot C_\lambda + B_\lambda \cdot h([\![C]\!]^\sigma) + m_A. \tag{8.1}$$

If $A \notin M^\lambda$ but $\bot \notin \{[\![B]\!]^\sigma, [\![B]\!]^\lambda, [\![C]\!]^\sigma, [\![C]\!]^\lambda\}$ then we take the same definition but omit the summand m_A.

Let us explain the definition (8.1): we have

$$\begin{aligned} [\![A]\!]^\sigma + [\![A]\!]^\lambda &= [\![A]\!] \\ &= [\![B]\!] \cdot [\![C]\!] = ([\![B]\!]^\sigma + [\![B]\!]^\lambda) \cdot ([\![C]\!]^\sigma + [\![C]\!]^\lambda) \\ &= [\![B]\!]^\lambda \cdot [\![C]\!]^\lambda + [\![B]\!]^\sigma \cdot [\![C]\!]^\lambda + [\![B]\!]^\lambda \cdot [\![C]\!]^\sigma + [\![B]\!]^\sigma \cdot [\![C]\!]^\sigma. \end{aligned}$$

By selecting from the last line all monomials of length at least n, we get

$$[\![A]\!]^\lambda = [\![B]\!]^\lambda \cdot [\![C]\!]^\lambda + [\![B]\!]^\sigma \cdot [\![C]\!]^\lambda + [\![B]\!]^\lambda \cdot [\![C]\!]^\sigma + m_A.$$

Applying to this equality the morphism h shows that (8.1) is indeed the right definition for $\mathsf{rhs}_\lambda(A_\lambda)$.

Case 3: $\mathsf{rhs}(A) = B \cdot C$ and $\bot \in \{[\![B]\!]^\sigma, [\![B]\!]^\lambda, [\![C]\!]^\sigma, [\![C]\!]^\lambda\}$. Then the corresponding terms on the right-hand side of (8.1) are omitted. More precisely if $[\![X]\!]^\sigma = \bot$ ($X \in \{B, C\}$), then we omit in (8.1) the product involving $h([\![X]\!]^\sigma)$ as well as the semiring element m_A and if $[\![X]\!]^\lambda = \bot$ ($X \in \{B, C\}$), then we omit in (8.1) the two products involving X_λ. The reader may also interpret $\bot$ as zero and then do the obvious simplifications in (8.1) (but note that the semiring $\mathbb{N}[R]$ has no additive zero element). For example, if $[\![B]\!]^\lambda = [\![C]\!]^\sigma = \bot$ and $[\![B]\!]^\sigma \neq \bot \neq [\![C]\!]^\lambda$, then $\mathsf{rhs}_\lambda(A_\lambda) = h([\![B]\!]^\sigma)C_\lambda$. Since $[\![A]\!]^\lambda \neq \bot$, at least one of the summands in (8.1) remains.

Figure 8.2 shows an example of the above construction of the circuit $\mathcal{C}^\lambda$, where $n = 2$. The shaded parts are those parts that are removed because they would yield zero terms. These are exactly those parts of the right-hand sides that are removed in the above Case 3. To the right of each gate X, the value $[\![X]\!]^\sigma$ is written. Note that the output gate evaluates to $2aab + 2ab$ which is indeed $(2aab + 2ab + 2a)^\lambda$.

Step 5. We now apply Lemma 5.3 and transform in logarithmic space $\mathcal{C}_\lambda$ into a circuit $\mathcal{C}'$ in normal form. To eliminate the remaining inputs from $R \setminus \langle R^n \rangle$ we have to argue that the circuit $\mathcal{C}'$ satisfies the conditions from Lemma 8.8 for the ideal $I = \langle R^n \rangle$. The input values of the circuit $\mathcal{C}_\lambda$ are elements from $\langle R^n \rangle$ (they occur as the m_A in (8.1)) and the $h([\![X]\!]^\sigma)$ for $X \in V$. Only the input

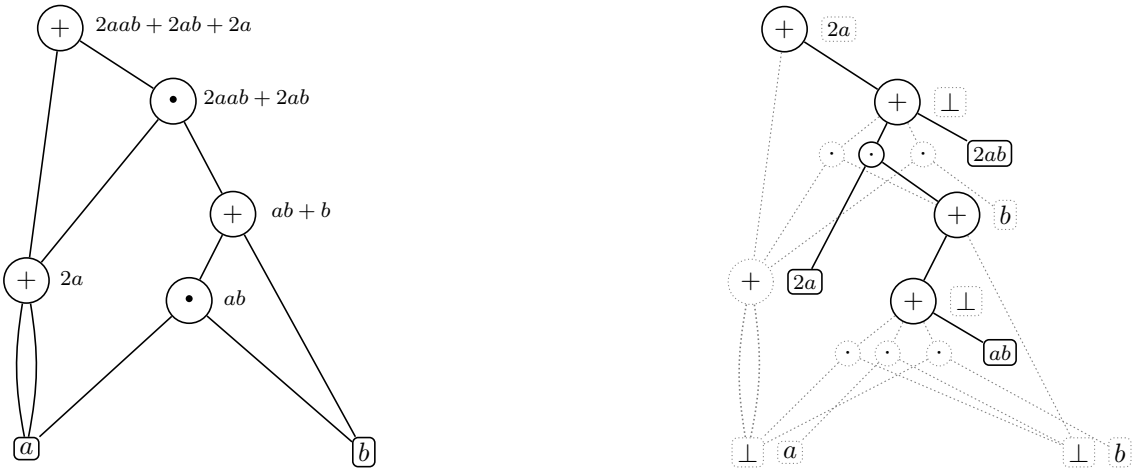


Figure 8.2: The construction of the circuit $\mathcal{C}_\lambda$ in the proof of Lemma 8.10.

values $h([\![X]\!]^\sigma)$ can belong to $R \setminus \langle R^n \rangle$ (they can also belong to $\langle R^n \rangle$). The circuit $\mathcal{C}'$ is obtained from $\mathcal{C}_\lambda$ by (i) eliminating copy gates (that may arise from the above Case 1) and (ii) splitting up right-hand sides of the form (8.1) (or a simpler form, see Case 3). Note that in the circuit $\mathcal{C}'$ an input gate Z with $\mathsf{rhs}_{\mathcal{C}'}(Z) \in R \setminus \langle R^n \rangle$ can only occur in right-hand sides of multiplication gates. Such a right-hand side must be of the form $B_\lambda \cdot Z$ or $Z \cdot C_\lambda$ (which is obtained from splitting up the expression in (8.1)). Here, B_λ and C_λ are gates from the circuit $\mathcal{C}_\lambda$. But a gate A_λ of the circuit $\mathcal{C}_\lambda$ cannot be transformed into an input gate of $\mathcal{C}'$ with a right-hand side from $R \setminus \langle R^n \rangle$ (note that the values $h([\![X]\!]^\sigma)$ are "guarded" in (8.1) by multiplications with gates Y_λ). This shows that the conditions for Lemma 8.8 are satisfied.

Finally, Lemma 8.8 allows us to transform in logarithmic space the circuit $\mathcal{C}'$ into an equivalent circuit $\mathcal{D}$ over the subsemiring $\langle R^n \rangle = \langle RER \rangle$. This circuit $\mathcal{D}$ satisfies $[\mathcal{D}] = h([\![\mathcal{C}]\!]^\lambda)$. We output this circuit together with the previously computed value $h([\![\mathcal{C}]\!]^\sigma)$ (if this value is not $\bot$). Then the output specification from the lemma is satisfied. □

The next lemma transforms a circuit over $\langle RER \rangle$ into a type admitting circuit.

Lemma 8.11. *Given a circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over $\langle RER \rangle$, one can compute in $\mathsf{AC}^0(\mathsf{NL})$:*

- *a type admitting circuit $\mathcal{C}' = (V', \mathsf{rhs}')$ (without output gate),*
- *a non-empty list of distinguished gates $A_1, \ldots, A_m \in V'$, where $m \leq |R|^4$, and*
- *elements $\ell_1, r_1, \ldots, \ell_m, r_m \in R$ such that $[\mathcal{C}] = \sum_{i=1}^m \ell_i [A_i]_{\mathcal{C}'} r_i$.*

Proof. Let us interpret the circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over the free semiring $\mathbb{N}[R]$ again. For each input gate A we can write $[A]$ as $\sum_{i=1}^k s_i e_i^3 t_i$ for a constant k (that only depends on R), $s_i, t_i \in R$, $e_i \in E$ and redefine $\mathsf{rhs}(A) = \sum_{i=1}^k s_i e_i^3 t_i$ (a sum of k monomials of length 5). Thus for all $A \in V$ we have $\mathsf{supp}([\![A]\!]_{\mathcal{C}}) \subseteq (RE^3R)^+ \subseteq RER^*ER$.

Let us define for every inner gate A of $\mathcal{C}$ the set

$$P_A = \{(s, e, f, t) \in R \times E \times E \times R \mid \mathsf{supp}([\![A]\!]_{\mathcal{C}}) \cap seR^*ft \neq \emptyset\}.$$

Hence, $|P_A| \leq |R|^4$. We claim that the sets P_A can be computed in $\mathsf{AC}^0(\mathsf{NL})$. For this, note that $(s, e, f, t) \in P_A$ if and only if

1. $e = f$ and there exists an input gate C of $\mathcal{C}$ such that the following conditions hold:
   - $\mathsf{rhs}(C)$ contains the monomial se^3t.

- There is a path from C to A (possibly the empty path) where all gates are addition gates.

or

2. There exist input gates C_1, C_2 of $\mathcal{C}$, and a multiplication gate B such that the following conditions hold:
   - $\mathsf{rhs}(C_1)$ contains the monomial se^3t' for some $t' \in R$.
   - $\mathsf{rhs}(C_2)$ contains the monomial $s'f^3t$ for some $s' \in R$.
   - There is a path from C_1 to B such that for every edge (X, Y) along this path where Y is a multiplication gate, $\mathsf{rhs}(Y) = X \cdot Z$ for some gate Z.
   - There is a path from C_2 to B such that for every edge (X, Y) along this path where Y is a multiplication gate, $\mathsf{rhs}(Y) = Z \cdot X$ for some gate Z.
   - There is a path from B to A (possibly the empty path) where except for B all gates are addition gates.

These conditions can be checked in NL.

We next compute in logarithmic space a new circuit $\mathcal{D}$ that contains for every gate A of $\mathcal{C}$ and every tuple $(s, e, f, t) \in P_A$ a gate $A_{s,e,f,t}$ such that the following holds, where as usual $[\![A_{s,e,f,t}]\!]_{\mathcal{D}}$ denotes the evaluation of the gate $A_{s,e,f,t}$ in the free semiring $\mathbb{N}[R]$ and $L(A, s, e, f, t) := \mathsf{supp}([\![A]\!]_{\mathcal{C}}) \cap seR^*ft$ (which is non-empty since $(s, e, f, t) \in P_A$):

$$[\![A_{s,e,f,t}]\!]_{\mathcal{D}} = \sum_{w \in L(A,s,e,f,t)} [\![A]\!]_{\mathcal{C}}(w) \cdot w \tag{8.2}$$

Intuitively, we decompose the polynomial $[\![A]\!]_{\mathcal{C}}$ into several summands according to the first two and last two symbols in every monomial. We define the rules of $\mathcal{D}$ as follows, where A is a gate of $\mathcal{C}$:

Case 1. $\mathsf{rhs}(A) = \sum_{i=1}^{k} s_i e_i^3 t_i$. Then, we have $P_A = \{(s_i, e_i, e_i, t_i) \mid 1 \leq i \leq k\}$ and we set

$$\mathsf{rhs}(A_{s_i,e_i,e_i,t_i}) = s_i e_i^3 t_i.$$

Case 2. $\mathsf{rhs}(A) = B \cdot C$ and $(s, e, f, t) \in P_A$. We set

$$\mathsf{rhs}(A_{s,e,f,t}) = \sum_{(s,e,f',t') \in P_B} \sum_{(s',e',f,t) \in P_C} B_{s,e,f',t'} \cdot C_{s',e',f,t}.$$

Case 3. $\mathsf{rhs}(A) = B + C$ and $(s, e, f, t) \in P_A$. We set

$$\mathsf{rhs}(A_{s,e,f,t}) = B_{s,e,f,t} + C_{s,e,f,t}.$$

With these right-hand sides, property (8.2) is easy to verify.

The idea of the last step is the following: let $\overline{u} = (s, e, f, t) \in P_A$. Every non-commutative polynomial $[\![A_{\overline{u}}]\!]_{\mathcal{D}}$ has the property that each of its monomials starts with see and ends with fft. By factoring out the common prefix se and suffix ft, respectively, we can write $[\![A_{\overline{u}}]\!]_{\mathcal{D}} = segft$ where $g \in e\,\mathbb{N}[R]f$ or $g = e$ (the latter case occurs if A is an input gate with right-hand side se^3t, in which case we have $e = f$). We now construct in logarithmic space a circuit $\mathcal{C}'$, which contains gates $A'_{s,e,f,t}$ (where $A_{s,e,f,t}$ is a gate of $\mathcal{D}$ as above) such that in the free semiring $\mathbb{N}[R]$, $A'_{s,e,f,t}$ evaluates to the above polynomial g. We define the right-hand side of $A'_{s,e,f,t}$ again by a case distinction, where we use the right-hand sides for $\mathcal{D}$ that we defined in the Cases 1-3 above.

Case 1. $e = f$ and $\mathsf{rhs}(A_{s,e,e,t}) = se^3t$. Then, we set $\mathsf{rhs}(A'_{s,e,e,t}) = e$.

Case 2. $\mathsf{rhs}(A_{s,e,f,t}) = \sum_{(s,e,f',t') \in P_B} \sum_{(s',e',f,t) \in P_C} B_{s,e,f',t'} \cdot C_{s',e',f,t}$. We set

$$\mathsf{rhs}(A'_{s,e,f,t}) = \sum_{(s,e,f',t') \in P_B} \sum_{(s',e',f,t) \in P_C} B'_{s,e,f',t'}(f't's'e')C_{s',e',f,t}. \tag{8.3}$$

Case 3. $\mathsf{rhs}(A_{s,e,f,t}) = B_{s,e,f,t} + C_{s,e,f,t}$. We set

$$\mathsf{rhs}(A'_{s,e,f,t}) = B'_{s,e,f,t} + C'_{s,e,f,t}.$$

It is now straightforward to verify that for every gate A of $\mathcal{C}$ we have:

$$[\![A]\!]_{\mathcal{C}} = \sum_{(s,e,f,t)\in P_A} [\![A_{s,e,f,t}]\!]_{\mathcal{D}} = \sum_{(s,e,f,t)\in P_A} se[\![A'_{s,e,f,t}]\!]_{\mathcal{C}'} ft.$$

Hence, if we evaluate the circuits $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{C}'$ in the semiring R we get

$$[A]_{\mathcal{C}} = \sum_{(s,e,f,t)\in P_A} [A_{s,e,f,t}]_{\mathcal{D}} \quad = \quad \sum_{(s,e,f,t)\in P_A} se[A'_{s,e,f,t}]_{\mathcal{C}'} ft.$$

Note that $[A'_{s,e,f,t}]_{\mathcal{C}'} \in eRf$, which holds, since $[A'_{s,e,f,t}]_{\mathcal{C}'} = h([\![A'_{s,e,f,t}]\!]_{\mathcal{C}'})$ and every monomial of $[\![A'_{s,e,f,t}]\!]_{\mathcal{C}'}$ starts with e and ends with f. Moreover, every input value of the circuit $\mathcal{C}'$ is from ERE: these are the elements e (in Case 1) and $f't's'e'$ (in Case 2).

Note that $\mathcal{C}'$ admits a type function: we set $\mathsf{type}(A'_{s,e,f,t}) = (e, f)$. Moreover, using Lemma 5.3 we transform $\mathcal{C}'$ in logarithmic space into normal form by splitting up right-hand sides of the form (8.3). Thereby we extend the type-mapping to the new gates that are introduced. For instance, if we introduce a gate with right-hand side $B'_{s,e,f',t'}(f't's'e')$ (which occurs in (8.3)), then this gate gets the type (e, e'), and the gate that computes (in two steps) $B'_{s,e,f',t'}(f't's'e')C_{s',e',f,t}$ gets the type (e, f). This ensures that the three conditions from Definition 8.4 are satisfied. □

Combining Lemma 8.10 and 8.11 immediately yields Lemma 8.6.

8.2.2 Step 2: a parallel evaluation algorithm for type admitting circuits

In this section we prove Lemma 8.7 by presenting a parallel evaluation algorithm for type admitting circuits. This algorithm terminates after at most $|R|$ rounds, if R has a so-called rank-function, which we define first. As before, let $E = E(R)$.

Definition 8.12 (rank-function)**.** *We call a function* $\mathsf{rank} : R \to \mathbb{N} \setminus \{0\}$ *a rank-function for* R *if it satisfies the following conditions for all* $a, b \in R$*:*

1. $\mathsf{rank}(a) \leq \mathsf{rank}(a + b)$
2. $\mathsf{rank}(a), \mathsf{rank}(b) \leq \mathsf{rank}(a \cdot b)$
3. *If* $a, b \in eRf$ *for some* $e, f \in E$ *and* $\mathsf{rank}(a) = \mathsf{rank}(a + b)$*, then* $a = a + b$*.*

Note that if $\mathcal{R}^\bullet$ is a monoid, then one can choose $e = 1 = f$ in the third condition in Definition 8.12, which is therefore equivalent to: if $\mathsf{rank}(a) = \mathsf{rank}(a + b)$ for $a, b \in R$, then $a = a + b$.

Example 8.13. *Let* $(G, \cdot)$ *be a finite group and consider the semiring* $\mathcal{P}(G) := (2^G \setminus \emptyset, \cup, \cdot)$ *with* $A \cdot B := \{a \cdot b \mid a \in A, b \in B\}$ *(In the next chapter we take a closer look at these so-called power semiring of* G*). One can verify that the function* $A \mapsto |A|$ *where* $\emptyset \neq A \subseteq G$*, is a rank-function for* $\mathcal{P}(G)$*. On the other hand, if* S *is a finite semigroup, which is not a group, then* S *cannot be cancellative. Assume that* $ab = ac$ *for* $a, b, c \in S$ *with* $b \neq c$*. Then* $\{a\} \cdot \{b, c\} = \{ab\}$*. This shows that the function* $A \mapsto |A|$ *is not a rank-function for* $\mathcal{P}(S)$*.*

Theorem 8.14. *If the finite semiring* R *has a rank-function* rank*, then the restriction of* $\mathsf{CEP}(R)$ *to type admitting circuits belongs to* $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(\mathcal{R}^+), \mathsf{CEP}(\mathcal{R}^\bullet))$*.*

Proof. Let $\mathcal{C} = (V, S, \mathsf{rhs})$ be a type admitting circuit with the rank-function rank. We present an algorithm which partially evaluates the circuit in a constant number of phases, where each phase can be carried out in $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(\mathcal{R}^+), \mathsf{CEP}(\mathcal{R}^\bullet))$ and the following invariant is preserved:

Invariant: after phase k all gates A with $\mathsf{rank}([A]_{\mathcal{C}}) \leq k$ are evaluated, i.e., are input gates in phase $k+1$ onwards.

Initially, i.e., for $k = 0$, the invariant holds, since 0 is not in the range of the rank-function. After $\max\{\mathsf{rank}(a) \mid a \in R\}$ (which is a constant) many phases, the output gate S is evaluated. We present phase k of the algorithm, assuming that the invariant holds after phase $k-1$. Thus, all gates A with $\mathsf{rank}([A]_{\mathcal{C}}) < k$ of the current circuit $\mathcal{C}$ are input gates. In phase k we evaluate all gates A with $\mathsf{rank}([A]_{\mathcal{C}}) = k$. For this, we proceed in two steps:

Step 1. As a first step the algorithm evaluates all subcircuits that only contain addition and input gates. This maintains the invariant and is possible in $\mathsf{AC}^0(\mathsf{NL}, \mathsf{CEP}(\mathcal{R}^+))$. After this step, every addition-gate A has at least one inner input gate, which we denote by $\mathsf{inner}(A)$ (if both input gates are inner gates, then choose one arbitrarily). The NL-oracle access is needed to compute the set of all gates A for which no multiplication gate $B \leq_{\mathcal{C}} A$ exists.

Step 2. Define the multiplicative circuit $\mathcal{C}' = (V, S, \mathsf{rhs}')$ by

$$\mathsf{rhs}'(A) = \begin{cases} \mathsf{inner}(A) & \text{if } A \text{ is an addition-gate,} \\ \mathsf{rhs}(A) & \text{if } A \text{ is a multiplication gate or an input gate.} \end{cases} \tag{8.4}$$

The circuit $\mathcal{C}'$ can be brought in logarithmic space into normal form by Lemma 5.3 and then evaluated in $\mathsf{AC}^0(\mathsf{CEP}(\mathcal{R}^\bullet))$. A gate $A \in V$ is called *locally correct* if (i) A is an input gate or multiplication gate of $\mathcal{C}$, or (ii) A is an addition gate of $\mathcal{C}$ with $\mathsf{rhs}(A) = B + C$ and $[A]_{\mathcal{C}'} = [B]_{\mathcal{C}'} + [C]_{\mathcal{C}'}$. We compute the set $W := \{A \in V \mid \text{ all gates } B \text{ with } B \leq_{\mathcal{C}} A \text{ are locally correct}\}$ in $\mathsf{AC}^0(\mathsf{NL})$. A simple induction shows that for all $A \in W$ we have $[A]_{\mathcal{C}} = [A]_{\mathcal{C}'}$. Hence we can set $\mathsf{rhs}(A) = [A]_{\mathcal{C}'}$ for all $A \in W$. This concludes phase k of the algorithm.

To prove that the invariant holds after phase k, we show that for each gate $A \in V$ with $\mathsf{rank}([A]_{\mathcal{C}}) \leq k$ we have $A \in W$. This is shown by induction: assume that $\mathsf{rank}([A]_{\mathcal{C}}) \leq k$. By the first condition from Definition 8.12, all gates $B <_{\mathcal{C}} A$ satisfy $\mathsf{rank}([B]_{\mathcal{C}}) \leq k$. Thus, the induction hypothesis yields $B \in W$ and hence $[B]_{\mathcal{C}} = [B]_{\mathcal{C}'}$ for all gates $B <_{\mathcal{C}} A$. It remains to show that A is locally correct, which is clear if A is an input gate or a multiplication gate. So assume that $\mathsf{rhs}(A) = B + C$ where $B = \mathsf{inner}(A)$, which implies $[A]_{\mathcal{C}'} = [B]_{\mathcal{C}'}$ by (8.4). Since B is an inner gate, which is not evaluated after phase $k-1$, it holds that $\mathsf{rank}([B]_{\mathcal{C}}) \geq k$ and therefore $\mathsf{rank}([B + C]_{\mathcal{C}}) = \mathsf{rank}([A]_{\mathcal{C}}) = \mathsf{rank}([B]_{\mathcal{C}}) = k$. Since $\mathcal{C}$ is type admitting, by Definition 8.4 there exist a type-function and idempotents $e, f \in E$ with $\mathsf{type}(B) = \mathsf{type}(C) = (e, f)$ and thus $[B]_{\mathcal{C}}, [C]_{\mathcal{C}} \in eRf$. The second condition from Definition 8.12 implies that $[A]_{\mathcal{C}} = [B]_{\mathcal{C}} + [C]_{\mathcal{C}} = [B]_{\mathcal{C}}$. We finally get $[A]_{\mathcal{C}'} = [B]_{\mathcal{C}'} = [B]_{\mathcal{C}} = [A]_{\mathcal{C}} = [B]_{\mathcal{C}} + [C]_{\mathcal{C}} = [B]_{\mathcal{C}'} + [C]_{\mathcal{C}'}$. Therefore A is locally correct. □

Example 8.15 (Example 8.13 continued)**.** *Figure 8.3 shows a circuit $\mathcal{C}$ over the power semiring $\mathcal{P}(G)$ of the group $G = (\mathbb{Z}_5, +)$. Recall from Example 8.13 that the function $A \mapsto |A|$ is a rank function for $\mathcal{P}(G)$. We illustrate one phase of the algorithm. All gates A with $\mathsf{rank}([A]) < 3$ are evaluated in the circuit $\mathcal{C}$ shown in (a). The goal is to evaluate all gates A with $\mathsf{rank}([A]) = 3$. The circuit $\mathcal{C}'$ (shown in (b)) from the proof of Theorem 8.14 is computed and evaluated using the oracle for $\mathsf{CEP}(\mathbb{Z}_5, +)$. The dotted wires do not belong to the circuit $\mathcal{C}'$. All locally correct gates are shaded. Note that the output gate is locally correct but its right child is not locally correct. All other shaded gates form a downwards closed set, which is the set W from the proof. These gates can be evaluated such that in the resulting circuit (shown in (c)) all gates which evaluate to elements of rank 3 are evaluated.*

For the proof of Lemma 8.7, it remains to show that every finite $\{0,1\}$-free semiring has a rank-function.

Lemma 8.16. *Let R be $\{0,1\}$-free. If $e, f \in E$ and $f = ef = fe = f + f$, then $e + f = f$.*

Proof. With $f = 0$ and $e + f = 1$ all equations from Lemma 3.25 (point 4) hold. Hence, we must have $e + f = f$. □

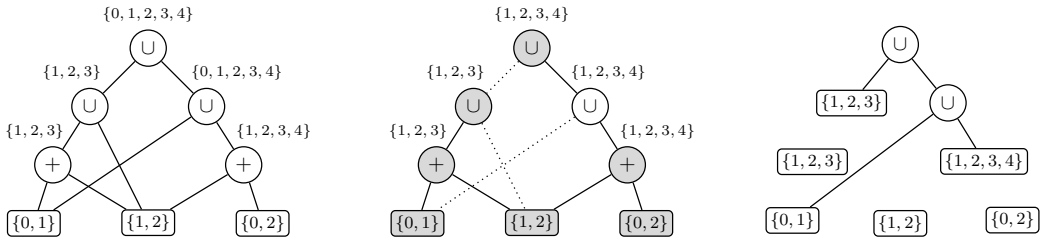


Figure 8.3: The parallel evaluation algorithm over the power semiring $\mathcal{P}(\mathbb{Z}_5)$.

Lemma 8.17. *If the finite semiring R is $\{0,1\}$-free, then R has a rank-function.*

Proof. For $a, b \in R$ we define $a \preceq b$ if b can be obtained from a by iterated additions and left- and right-multiplications of elements from R. This is equivalent to the following condition:

there are $\ell, r, c \in R$ such that $b = \ell a r + c$ (where each of the elements ℓ, r, c can be also missing).

Since $\preceq$ is a preorder on R, there is a function $\mathsf{rank} : R \to \mathbb{N} \setminus \{0\}$ such that for all $a, b \in R$ we have

- $\mathsf{rank}(a) = \mathsf{rank}(b)$ exactly if $a \preceq b$ and $b \preceq a$,
- $\mathsf{rank}(a) \leq \mathsf{rank}(b)$ if $a \preceq b$.

We claim that rank satisfies the conditions of Definition 8.12. The first two conditions are clear, since $a \preceq a + b$ and $a, b \preceq ab$. For the third condition, let $e, f \in E$ and $a, b \in eRf$ such that $\mathsf{rank}(a+b) = \mathsf{rank}(a)$, which is equivalent to $a+b \preceq a$. Assume that $a = \ell(a+b)r+c = \ell a r+\ell b r+c$ for some $\ell, r, c \in R$ (the case without c can be handled in the same way). Since $a = eaf$ and $b = ebf$, we have $a = \ell e(a+b)fr + c$ and hence we can assume that ℓ and r are not missing. By multiplying with e from the left and f from the right we get $a = (e\ell e)(a+b)(frf) + (ecf)$, so we can assume that $\ell = e\ell e$ and $r = frf$. After m repeated applications of $a = \ell a r + \ell b r + c$ we obtain

$$a = \ell^m a r^m + \sum_{i=1}^{m} \ell^i b r^i + \sum_{i=0}^{m-1} \ell^i c r^i. \tag{8.5}$$

Let $n \geq 1$ such that nx is additively idempotent and x^n is multiplicatively idempotent for all $x \in R$. Hence nx^n is both additively and multiplicatively idempotent for all $x \in R$. If we choose $m = n^2$, the right-hand side of (8.5) contains the partial sum $\sum_{i=1}^{n} \ell^{in} b r^{in}$. Furthermore, $e(n\ell^n) = (n\ell^n)e = n\ell^n$ and $f(nr^n) = (nr^n)f = nr^n$. Therefore, Lemma 8.16 implies that $n\ell^n = n\ell^n + e$ and $nr^n = nr^n + f$, and hence:

$$\begin{aligned}
\sum_{i=1}^{n} \ell^{in} b r^{in} &= n(\ell^n b r^n) = n^2(\ell^n b r^n) = (n\ell^n)b(nr^n) = (n\ell^n + e)b(nr^n) \\
&= (n\ell^n)b(nr^n) + eb(nr^n) = (n\ell^n)b(nr^n) + eb(nr^n + f) \\
&= (n\ell^n)b(nr^n) + eb(nr^n) + ebf = \left(\sum_{i=1}^{n} \ell^{in} b r^{in}\right) + b.
\end{aligned}$$

Thus, we can replace in (8.5) the partial sum $\sum_{i=1}^{n} \ell^{in} b r^{in}$ by $\sum_{i=1}^{n} \ell^{in} b r^{in} + b$, which proves that $a = a + b$. □

Chapter 9

The circuit intersection problem for semigroups

9.1 Introduction

In 1973 Stockmeyer and Meyer [80] extended the word problem for some algebraic structures to the following problem over the power set of these structures where the operations on sets are evaluated via $A \cdot B = \{a \cdot b \mid a \in A, b \in B\}$ and the union is added as an additional operation: for a given expression over this structure and an element a, decide whether a is an element of the set to which the expression evaluates. For instance, Stockmeyer and Meyer showed that this problem over $(2^{\mathbb{Z}}, +, \cup)$ is NP-complete. In this chapter we will extend the circuit value problem in a similar way. We consider the power set of the domain of an algebraic structure without the empty set with the operations as defined above and the union. For two circuits over this power structure we ask whether the intersection of the represented sets is non-empty. We call this problem the circuit intersection problem, or short "circuit intersection". In [62] resp. [84] the authors include even more operations and consider circuits over reducts (i.e., structures where some of the operations are omitted) of the power set structures $(2^{\mathbb{N}}, +, \cdot, \cup, \cap, ^-)$ resp., $(2^{\mathbb{Z}}, +, \cdot, \cup, \cap, ^-)$ (here $^-$ stands for the complement of a set). In particular, they considered the circuit intersection problem for circuits over $(2^{\mathbb{Z}}, +, \cdot, \cup)$ (NEXPTIME-complete), over $(2^{\mathbb{N}}, +, \cdot, \cup)$ (PSPACE-complete), and over $(2^{\mathbb{Z}}, +, \cup)$ (resp. $(2^{\mathbb{N}}, +, \cup)$) (both NP-complete). Notice that they did not exclude the empty set explicitly, but since their inputs are restricted to singletons, gates from circuits over $(2^{\mathbb{N}}, +, \cdot, \cup)$ resp. $(2^{\mathbb{Z}}, +, \cdot, \cup)$ can never evaluate to the empty set. In the following we consider the circuit intersection problem for two structures: first we use the results about the circuit evaluation problem for finite semirings to show a dichotomy for the circuit intersection problem for finite semigroups: in the case S is a finite local group the circuit intersection problem for S can be reduced to circuit evaluation for S. In all other cases the power structure contains a $\{0, 1\}$-subsemiring and the circuit intersection problem is P-complete. Then we show that for the group $\mathsf{SL}_5(\mathbb{Z})$ circuit intersection is NEXPTIME-complete. The result for finite semigroups was shown in [3].

9.2 The circuit intersection problem

Definition 9.1 (power structure)**.** *Let $\mathcal{A} = (D, f_1, \ldots, f_k)$ be an algebraic structure. Then $\mathcal{P}(\mathcal{A}) = (2^D \setminus \emptyset, f_1, \ldots, f_k)$ is the power structure of $\mathcal{A}$ where $f_i : (2^D \setminus \emptyset)^{n_i} \to (2^D \setminus \emptyset)$ is defined by $f_i(M_1, \ldots, M_{n_i}) = \{f_i(m_1, \ldots, m_{n_i}) \mid m_j \in M_j \text{ for } 1 \leq j \leq n_i\}$.*

Notice that it makes sense to exclude the empty set in the definition above, since if I is a generating set of $\mathcal{A}$, then the one-element subsets of I form a generating set of $\mathcal{P}(\mathcal{A})$. So the operations of the power structure of $\mathcal{A}$ can be seen as simultaneously acting on the elements

of $\mathcal{A}$. If we allowed the empty set in the domain of $\mathcal{P}(\mathcal{A})$, this would lead to the fact that $f_i(M_1, \ldots, \emptyset, \ldots, M_{n_i}) = \emptyset$ for all $M_1, \ldots, M_{n_i} \in 2^D$ which would in some sense destroy the natural structure given by $\mathcal{A}$.

In this chapter we extend our considerations to circuits over power structures. For these circuits we investigate the following problem. The circuit intersection problem for the algebraic structure $\mathcal{A}$ (short $\mathsf{CIP}(\mathcal{A})$) is the following:

Input: Two circuits $\mathcal{C}$ and $\mathcal{D}$ over $\mathcal{P}(\mathcal{A})$
Question: Is $[\mathcal{C}] \cap [\mathcal{D}] \neq \emptyset$?

As for the circuit evaluation problem for unitary rings, groups and finite structures a slightly different but equivalent version of the problem is used:

- For a unitary ring $(R, +, \cdot)$ we consider the following problem:

  Input: A circuit $\mathcal{C}$ over $\mathcal{P}(R)$.
  Question: Is $0 \in [\mathcal{C}]$?

- For a group $(G, \cdot)$ we consider the following problem:

  Input: A circuit $\mathcal{C}$ over $\mathcal{P}(G)$.
  Question: Is $1 \in [\mathcal{C}]$?

- For a finite algebraic structure $\mathcal{A} = (D, f_1, \ldots, f_k)$ we consider the following problem:

  Input: A circuit $\mathcal{C} = (V, S, \mathsf{rhs})$ over $\mathcal{P}(\mathcal{A})$ and an element $a \in D$.
  Question: Is $a \in [\mathcal{C}]$?

Note that the problems above are AC^0-equivalent to the circuit intersection problem and that the equivalence can be shown in the same way as it was done for the circuit evaluation problem in Section 5.2.

9.3 The circuit intersection problem for finite semigroups

Definition 9.2 (local group)**.** *A semigroup S is called local group if for all $e \in E(S)$ the local monoid eSe is a group.*

It is known that in every finite local group S of size n the minimal ideal of S is $S^n = SE(S)S$, see [8, Proposition 2.3].

Notice that the power structure of a finite semigroup is a finite semiring. So we can use Corollary 8.3 to show the following result:

Theorem 9.3. *Let S be a finite semigroup. If S is a local group and solvable, then $\mathsf{CIP}(S)$ belongs to DET. Otherwise $\mathsf{CIP}(S)$ is P-complete.*

Proof. First let S be a finite local group which is solvable, $a \in S$ and $\mathcal{D}$ a circuit over $\mathcal{P}(S)$. By [15, Corollary 2.7] the multiplicative semigroup $\mathcal{P}(S)^\bullet$ is solvable as well. We show that the semiring $\mathcal{P}(S)$ is $\{0,1\}$-free: towards a contradiction assume that $\mathcal{P}(S)$ is not $\{0,1\}$-free. By Lemma 3.25, there exist non-empty sets $A \subsetneq B \subseteq S$ such that $AB = BA = A^2 = A$ and $B^2 = B$. Hence, B is a subsemigroup of S, which is also a local group, and A is an ideal in B. Since the minimal ideal of B is B^n for $n = |B|$ and $B^n = B$, we obtain $A = B$, which is a contradiction. So we can test in parallel for every subset $A \subseteq S$ that contains a whether $A = [\mathcal{D}]$ in DET (notice that $|S|$ is a constant). So $\mathsf{CIP}(S)$ belongs to DET in this case. Now assume that S is not a local group, i.e., there exists a local monoid $M = eSe$ which is not a group for some idempotent $e \in S$. Since any finite monoid which is not a group contains two distinct idempotents (Lemma 3.4), there is an idempotent $f \in M$ such that $f \neq e$ and $\mathcal{P}(S)$ is not $\{0,1\}$-free: $\{f\}$ and $\{e, f\}$ form a copy of $\mathbb{B}_2$. It follows that the question whether $e \in [\mathcal{D}]$ is P-complete. So $\mathsf{CIP}(S)$ is P-complete. Finally, if S is not solvable, then also $\mathcal{P}(S)^\bullet$ is not solvable and $\mathsf{CIP}(S)$ is P-complete by Theorem 5.15. □

9.4 The circuit intersection problem for $\mathsf{SL}_5(\mathbb{Z})$

The circuit intersection problem for groups can be seen as a nondeterministic version of the compressed word problem [57]. In the previous section it was shown that for finite groups the circuit intersection problem is as hard as circuit evaluation for G. In contrast to this result, considering the power-semiring over an infinite group G we get a massive increase of complexity when we compare circuit intersection for G to $\mathsf{CEP}(G)$. For instance, in Section 5.3 we have seen that $\mathsf{CEP}(\mathbb{Z})$ is $\mathsf{C}_=\mathsf{L}$-complete, while in [62] and [84] it was shown that $\mathsf{CIP}(\mathbb{Z})$ is NP-complete. Using results from [54], one can show that the circuit intersection problem for F_2 where F_2 is the free group of rank 2, is PSPACE-complete, while $\mathsf{CEP}(F_2)$ can be solved in polynomial time [57]. Here we show a similar result for linear groups. From [57] we know that for every f.g. linear group G circuit evaluation can be reduced to polynomial identity testing over $\mathbb{Z}$ or $\mathbb{Z}_p$ for a prime p and hence is in coRP. For the linear group $\mathsf{SL}_5(\mathbb{Z})$ we show the following:

Theorem 9.4. *Circuit intersection for* $\mathsf{SL}_5(\mathbb{Z})$ *is* $\mathsf{NEXPTIME}$*-complete.*

First we show that $\mathsf{CIP}(\mathsf{SL}_5(\mathbb{Z}))$ is in $\mathsf{NEXPTIME}$ similarly to the proof in [84], where it was shown that the circuit intersection problem for $(\mathbb{Z},+,\cdot)$ is in $\mathsf{NEXPTIME}$: we can unfold a given circuit $\mathcal{C}$ in exponential time into a formula F (i.e., a circuit where the corresponding graph is a tree) that is possibly exponentially larger than $\mathcal{C}$. Then we replace nondeterministically every subexpression $A \cup B$ by either A or B and obtain a tree F' where only multiplication gates are left. Now we can check in polynomial time (in $|F'|$) whether $\mathsf{Id} = [F'] \in [\mathcal{C}]$. The harder part is to show that $\mathsf{CIP}(\mathsf{SL}_5(\mathbb{Z}))$ is $\mathsf{NEXPTIME}$-hard: to do this, we consider $\mathsf{CIP}(\mathbb{Z},+,\cdot)$ where the input sets are restricted to $\{-1\}$ and $\{1\}$ which is known to be NEXPTIME-complete [84]. We show in two steps that this problem can be reduced to $\mathsf{CIP}(\mathsf{SL}_5(\mathbb{Z}))$: first we show a reduction to $\mathsf{CIP}(\mathsf{SL}_5(\mathbb{Z}[x_1,\ldots,x_k]))$ where k is part of the input and then we show that the variables can be replaced by large integers with an idea from [6]. The transformation from a circuit $\mathcal{C}$ over $\mathcal{P}(\mathbb{Z},+,\cdot)$ into a circuit $\mathcal{D}$ over $\mathcal{P}(\mathsf{SL}_5(\mathbb{Z}[x_1,\ldots,x_k]))$ is done as follows: let $\mathcal{C} = (V, S, \mathsf{rhs})$ be a circuit over $\mathcal{P}(\mathbb{Z},+,\cdot)$ and $X = \{x_{A,i} \mid A \in V, 1 \leq i \leq 10\}$ be a set of variables. We define a circuit $\mathcal{D} = (V', S_{1,2,1}, \mathsf{rhs}')$ with multiplication and union gates over $\mathcal{P}(\mathsf{SL}_5(\mathbb{Z}[X]))$ as follows: the set of the new gates V' is defined as:

$$\begin{aligned} V' = &\{A_{i,j,d} \mid A \in V, 1 \leq i,j \leq 3 \text{ with } i \neq j, d \in \{-1,1\}\} \cup \\ &\{T_{A,k,d} \mid A \in V, 1 \leq k \leq 3, d \in \{-1,1\}\} \cup \\ &\{T(A)_{i,j,d,l} \mid A \in V, 1 \leq i,j \leq 3 \text{ with } i \neq j, d \in \{-1,1\}, 1 \leq l \leq 4\}. \end{aligned}$$

Recall that with $T_{i,j}$ we denote the matrix such that all entries in the main diagonal are 1, $T_{i,j}[i,j] = 1$ and all other entries are 0. The meaning of the gates in V' is the following: $A_{i,j,d}$-gates evaluate to the set $\{T_{i,j}^{da} \mid a \in [A]_{\mathcal{C}}\} \cup I_A$ where I_A is a (possibly empty) set of some invalid matrices, that are explained in detail later. We can basically use the methods of the reduction from $\mathsf{CEP}(\mathbb{Z},+,\cdot)$ to $\mathsf{CEP}(\mathsf{SL}_3(\mathbb{Z}))$ [57] to obtain this property. More exactly, by Lemma 3.37 we can simulate the multiplication of integers by a product of four matrices. Here the following problem arises: for instance, let $A_1 = \{1,2\}$ and $A_2 = \{3\}$. Then $A_3 := A_1 \cdot A_2 = \{3,6\}$. A corresponding set of matrices would be $A_1' = \{T_{2,3}^1, T_{2,3}^2\}$ and $A_2' = \{T_{1,2}^3\}$. Assigning the formula from Lemma 3.37 to this case leads to $\{T_{2,3}^{-1}, T_{2,3}^{-2}\} \cdot \{T_{1,2}^3\} \cdot \{T_{2,3}^1, T_{2,3}^2\} \cdot \{T_{1,2}^{-3}\} = \{T_{1,3}^3, T_{1,3}^6, T_{1,3}^3 T_{2,3}^{-1}, T_{1,3}^6 T_{2,3}^1\}$. Since the latter two matrices do not correspond to a value in A_3 we want to mark them as invalid. This will be done by a gate of the form $T_{A,k,d}$ that multiplies the values at position $(2,3)$ (that are equal to zero in our example if the matrix corresponds to a value in A_3) with a variable and add these product to column 4 of the matrix. To ensure that matrices that were once marked as invalid by some variable stay invalid we will multiply the 4th and 5th rows and columns with some unique new variable before multiplying them once more. This is done by gates of the form $T(A)_{i,j,d,l}$. In detail we define the right-hand side of $\mathcal{D}$ as follows:

- $$\mathsf{rhs}'(T_{A,k,1}) = \left\{T_{4,k}^{x_{A,9}} T_{k,4}^{-x_{A,10}}\right\}$$

  and

  $$\mathsf{rhs}'(T_{A,k,-1}) = \left\{T_{k,4}^{x_{A,10}} T_{4,k}^{-x_{A,9}}\right\}.$$

- If $\mathsf{rhs}(A) = B + C$ or $\mathsf{rhs}(A) = B \cdot C$, then

  $$\mathsf{rhs}'(T(B)_{i,j,d,l}) = T_{4,5}^{x_{A,l}} T_{5,4}^{x_{A,l+4}} B_{i,j,d} T_{5,4}^{-x_{A,l+4}} T_{4,5}^{-x_{A,l}}$$

  and

  $$\mathsf{rhs}'(T(C)_{i,j,d,l}) = T_{4,5}^{x_{A,l}} T_{5,4}^{x_{A,l+4}} C_{i,j,d} T_{5,4}^{-x_{A,l+4}} T_{4,5}^{-x_{A,l}}.$$

  (That means e.g. for $l = 1$ that for every matrix $M \in B_{i,j,d}$ (resp. $M \in C_{i,j,d}$) we add $x_{A,5}$ times the 4th row to the 5th row and $-x_{A,5}$ times the 5th column to the 4th column. Afterwards we add $x_{A,1}$ times the 5th row to the 4th row and $-x_{A,1}$ times the 4th column to the 5th column.)

- If $\mathsf{rhs}(A) = \{e\}$ with $e \in \{-1, 1\}$, then

  $$\mathsf{rhs}'(A_{i,j,d}) = \left\{T_{i,j}^{ed}\right\}.$$

- If $\mathsf{rhs}(A) = B \cup C$, then

  $$\mathsf{rhs}'(A_{i,j,d}) = B_{i,j,d} \cup C_{i,j,d}.$$

- If $\mathsf{rhs}(A) = B + C$, then

  $$\mathsf{rhs}'(A_{i,j,d}) = T(B)_{i,j,d,1} \cdot T(C)_{i,j,d,2}.$$

- If $\mathsf{rhs}(A) = B \cdot C$, then let $\{k\} = \{1, 2, 3\} \setminus \{i, j\}$ and set

  $$\mathsf{rhs}'(A_{i,j,1}) = T_{A,k,1} \cdot T(B)_{k,j,-1,1} \cdot T(C)_{i,k,1,2} \cdot T(B)_{k,j,1,3} \cdot T(C)_{i,k,-1,4} \cdot T_{A,k,-1}$$

  and

  $$\mathsf{rhs}'(A_{i,j,-1}) = T_{A,k,1} \cdot T(B)_{k,j,-1,1} \cdot T(C)_{i,k,-1,2} \cdot T(B)_{k,j,1,3} \cdot T(C)_{i,k,1,4} \cdot T_{A,k,-1}.$$

  (If M is a matrix in $T(B)_{k,j,-1,1} \cdot T(C)_{i,k,-1,2} \cdot T(B)_{k,j,1,3} \cdot T(C)_{i,k,1,4}$, the two outer operations add $-x_{A,10}$ times the 4th row to the kth row and $x_{A,10}$ times the kth column to the 4th column. Afterwards they add $x_{A,9}$ times the kth row to the 4th row and $-x_{A,9}$ times the 4th column to the kth column.)

Finally let $S_{1,2,1}$ be the output gate of $\mathcal{D}$.

The definitions of $\mathsf{rhs}'(T(A)_{i,j,d,l})$ and $\mathsf{rhs}'(T_{A,k,d})$ lead immediately to the following result:

Lemma 9.5. *Let $i, j \in \{1, 2, 3\}$ and $i \neq j$.*

1. *If $T_{i,j}^e \in [A_{i,j,d}]_{\mathcal{D}}$ for some $e \in \mathbb{Z}$ and $\mathsf{rhs}'(T(A)_{i,j,d,l})$ is defined, then $T_{i,j}^e \in [T(A)_{i,j,d,l}]_{\mathcal{D}}$ for every $1 \leq l \leq 4$.*
2. *For $k \in \{1, 2, 3\} \setminus \{i, j\}$ and $e \in \mathbb{Z}$ holds $[T_{A,k,1}]_{\mathcal{D}} \{T_{i,j}^e\} [T_{A,k,-1}]_{\mathcal{D}} = \{T_{i,j}^e\}$.*

Lemma 9.6. *For the defined circuits $\mathcal{C}$ and $\mathcal{D}$ the following holds:*

$$\mathsf{Id} \in [\mathcal{D}] \text{ exactly if } 0 \in [\mathcal{C}].$$

Proof. We first show by induction that for every gate $A \in V$ and for every $a \in [A]_{\mathcal{C}}$ we get $T_{i,j}^{da} \in [A_{i,j,d}]_{\mathcal{D}}$ for every $1 \leq i,j \leq 3$ with $i \neq j$. So if $0 \in [\mathcal{C}]$, then $T_{1,2}^{0} = \mathsf{Id} \in [S_{1,2,1}]_{\mathcal{D}} = [\mathcal{D}]$. We only show the case $d = 1$, since for $d = -1$ exactly the same arguments hold. From now on fix a pair (i,j) with $1 \leq i,j \leq 3$ and $i \neq j$.

1. If A is an input-gate, the assumption holds by definition.
2. If the assumption holds for B and C and $\mathsf{rhs}(A) = B \cup C$, then $a \in [A]_{\mathcal{C}}$ implies $a \in [B]_{\mathcal{C}}$ or $a \in [C]_{\mathcal{C}}$. So $T_{i,j}^{a} \in [B_{i,j,1}]_{\mathcal{D}}$ or $T_{i,j}^{a} \in [C_{i,j,1}]_{\mathcal{D}}$. That means $T_{i,j}^{a} \in [B_{i,j,1}]_{\mathcal{D}} \cup [C_{i,j,1}]_{\mathcal{D}} = [A_{i,j,1}]_{\mathcal{D}}$.
3. If the assumption holds for B and C and $\mathsf{rhs}(A) = B + C$, then for every $a \in [A]_{\mathcal{C}}$ there are $b \in [B]_{\mathcal{C}}$ and $c \in [C]_{\mathcal{C}}$ with $a = b + c$. We know that $T_{i,j}^{b} \in [B_{i,j,1}]_{\mathcal{D}}$ and $T_{i,j}^{c} \in [C_{i,j,1}]_{\mathcal{D}}$. By Lemma 9.5 we know that also $T_{i,j}^{b} \in [T(B)_{i,j,1,1}]_{\mathcal{D}}$ and $T_{i,j}^{c} \in [T(C)_{i,j,1,2}]_{\mathcal{D}}$. So $T_{i,j}^{b} T_{i,j}^{c} = T_{i,j}^{b+c} = T_{i,j}^{a} \in [T(B)_{i,j,1,1}]_{\mathcal{D}} \cdot [T(C)_{i,j,1,2}]_{\mathcal{D}} = [A_{i,j,1}]_{\mathcal{D}}$.
4. If the assumption holds for B and C and $\mathsf{rhs}(A) = B \cdot C$, then for every $a \in [A]_{\mathcal{C}}$ there are $b \in [B]_{\mathcal{C}}$ and $c \in [C]_{\mathcal{C}}$ with $a = b \cdot c$ and we know for $k \in \{1,2,3\} \setminus \{i,j\}$ that $T_{k,j}^{db} \in [B_{k,j,d}]_{\mathcal{D}}$ and $T_{i,k}^{dc} \in [C_{i,k,d}]_{\mathcal{D}}$. By Lemma 9.5 we know that also $T_{k,j}^{-b} \in [T(B)_{k,j,-1,1}]_{\mathcal{D}}$, $T_{i,k}^{c} \in [T(C)_{i,k,1,2}]_{\mathcal{D}}$, $T_{k,j}^{b} \in [T(B)_{k,j,1,3}]_{\mathcal{D}}$ and $T_{i,k}^{-c} \in [T(C)_{i,k,-1,4}]_{\mathcal{D}}$. So

$$
\begin{aligned}
&\left(T_{4,k}^{x_{A,9}} T_{k,4}^{-x_{A,10}}\right) \left(T_{k,j}^{-b} T_{i,k}^{c}, T_{k,j}^{b} T_{i,k}^{-c}\right) \left(T_{k,4}^{x_{A,10}} T_{4,k}^{-x_{A,9}}\right) \\
&= \left(T_{4,k}^{x_{A,9}} T_{k,4}^{-x_{A,10}}\right) T_{i,j}^{bc} \left(T_{k,4}^{x_{A,10}} T_{4,k}^{-x_{A,9}}\right) \\
&= \left(T_{4,k}^{x_{A,9}} T_{k,4}^{-x_{A,10}}\right) T_{i,j}^{a} \left(T_{k,4}^{x_{A,10}} T_{4,k}^{-x_{A,9}}\right) \\
&\in [T_{A,k,1}]_{\mathcal{D}} [T(B)_{k,j,-1,1}]_{\mathcal{D}} [T(C)_{i,k,1,2}]_{\mathcal{D}} [T(B)_{k,j,1,3}]_{\mathcal{D}} [T(C)_{i,k,-1,4}]_{\mathcal{D}} [T_{A,k,-1}]_{\mathcal{D}} \\
&= [A_{i,j,1}]_{\mathcal{D}}.
\end{aligned}
$$

   By Lemma 9.5 we finally get that $T_{4,k}^{x_{A,9}} T_{k,4}^{-x_{A,10}} T_{i,j}^{a} T_{k,4}^{x_{A,10}} T_{4,k}^{-x_{A,9}} = T_{i,j}^{a} \in [A_{i,j,1}]_{\mathcal{D}}$.

So by induction this direction of the lemma holds.
Let $O = (\{1,2,3\} \times \{4,5\}) \cup (\{4,5\} \times \{1,2,3\})$ be the set of coordinates of the outer region of a matrix. To show the other direction of Lemma 9.6 we will show the following claim by induction: if $M \in [A_{i,j,1}]_{\mathcal{D}}$ and $M[x,y] = 0$ for every $(x,y) \in O$, then $M = T_{i,j}^{c}$ for some $c \in \mathbb{Z}$ and $1 \leq i,j \leq 3$ with $i \neq j$, and $c \in [A]_{\mathcal{C}}$. This especially implies that if $\mathsf{Id} = T_{i,j}^{0} \in [A_{i,j,1}]_{\mathcal{D}}$, then $0 \in [A]_{\mathcal{C}}$. Again the same arguments hold for $d = -1$.

1. If $A_{i,j,1}$ is an input gate, then the claim holds by definition.
2. If $rhs'(A_{i,j,1}) = B_{i,j,1} \cup C_{i,j,1}$ and $M \in [A_{i,j,1}]_{\mathcal{D}}$ with $M[x,y] = 0$ for $(x,y) \in O$, then $M \in [B_{i,j,1}]_{\mathcal{D}}$ or $M \in [C_{i,j,1}]_{\mathcal{D}}$ and so by our assumption $M = T_{i,j}^{c}$ for some $c \in \mathbb{Z}$ and $c \in [B]_{\mathcal{C}}$ or $c \in [C]_{\mathcal{C}}$. Hence $c \in [B]_{\mathcal{C}} \cup [C]_{\mathcal{C}} = [A]_{\mathcal{C}}$ and the claim holds.
3. If $\mathsf{rhs}'(A_{i,j,1}) = T(B)_{i,j,1,1} \cdot T(C)_{i,j,1,2}$ and $M \in [A_{i,j,1}]_{\mathcal{D}}$ with $M[x,y] = 0$ for $(x,y) \in O$, then there are matrices $M_1 \in [B_{i,j,1}]_{\mathcal{D}}$ and $M_2 \in [C_{i,j,1}]_{\mathcal{D}}$ such that

$$
M = \left(T_{4,5}^{x_{A,1}} T_{5,4}^{x_{A,5}}\right) M_1 \left(T_{5,4}^{-x_{A,5}} T_{4,5}^{-x_{A,1}}\right) \left(T_{4,5}^{x_{A,2}} T_{5,4}^{x_{A,6}}\right) M_2 \left(T_{5,4}^{-x_{A,6}} T_{4,5}^{-x_{A,2}}\right)
$$

   When we assume that $M_1[x,y] = p \neq 0$ for some $(x,y) \in O$, then by multiplication with the transformation matrices p is multiplied with $x_{A,1}$ or $x_{A,5}$ and added to another position $(x',y') \in O$. Since the variables $x_{A,1}$ and $x_{A,5}$ are used nowhere else, the value of $M[x',y']$ would not be equal to zero, which contradicts the assumption that $M[x,y] = 0$ for all $(x,y) \in O$. So we can assume that $M_1[x,y] = 0$ for all $(x,y) \in O$ and since the same

argumentation works for M_2 we can also assume that $M_2[x,y]=0$ for all $(x,y)\in O$. So by our induction assumption $M_1=T_{i,j}^b$ for some $b\in\mathbb{Z}$, $M_2=T_{i,j}^c$ for some $c\in\mathbb{Z}$, $b\in[B]_\mathcal{C}$ and $c\in[C]_\mathcal{C}$. So

$$M=\left(T_{4,5}^{x_{A,1}}T_{5,4}^{x_{A,5}}\right)T_{i,j}^b\left(T_{5,4}^{-x_{A,5}}T_{4,5}^{-x_{A,1}}\right)\left(T_{4,5}^{x_{A,2}}T_{5,4}^{x_{A,6}}\right)T_{i,j}^c\left(T_{5,4}^{-x_{A,6}}T_{4,5}^{-x_{A,2}}\right)=T_{i,j}^bT_{i,j}^c=T_{i,j}^{b+c}$$

and $b+c\in[B]_\mathcal{C}+[C]_\mathcal{C}=[A]_\mathcal{C}$.

4. If

$$\mathsf{rhs}'(A_{i,j,1})=T_{A,k,1}\cdot T(B)_{k,j,-1,1}\cdot T(C)_{i,k,1,2}\cdot T(B)_{k,j,1,3}\cdot T(C)_{i,k,-1,4}\cdot T_{A,k,-1}$$

with $k\in\{1,2,3\}\setminus\{i,j\}$ and $M\in[A_{i,j,1}]_\mathcal{D}$ with $M[x,y]=0$ for $(x,y)\in O$, then there are matrices $M_1\in[B_{k,j,-1}]_\mathcal{D}$, $M_2\in[C_{i,k,1}]_\mathcal{D}$, $M_3\in[B_{k,j,1}]_\mathcal{D}$ and $M_4\in[C_{i,k,-1}]_\mathcal{D}$ such that

$$\begin{aligned}M=\quad&\left(T_{4,k}^{x_{A,9}}T_{k,4}^{-x_{A,10}}\right)\left(T_{4,5}^{x_{A,1}}T_{5,4}^{x_{A,5}}\right)M_1\left(T_{5,4}^{-x_{A,5}}T_{4,5}^{-x_{A,1}}\right)\left(T_{4,5}^{x_{A,2}}T_{5,4}^{x_{A,6}}\right)M_2\left(T_{5,4}^{-x_{A,6}}T_{4,5}^{-x_{A,2}}\right)\\&\left(T_{4,5}^{x_{A,3}}T_{5,4}^{x_{A,7}}\right)M_3\left(T_{5,4}^{-x_{A,7}}T_{4,5}^{-x_{A,3}}\right)\left(T_{4,5}^{x_{A,4}}T_{5,4}^{x_{A,8}}\right)M_4\left(T_{5,4}^{-x_{A,8}}T_{4,5}^{-x_{A,4}}\right)\left(T_{k,4}^{x_{A,10}}T_{4,k}^{-x_{A,9}}\right).\end{aligned}$$

As argued in the previous case, if w.l.o.g. the matrix M_1 has an entry unequal 0 in O, then by the transformation matrices next to M_1 we get a multiple of the unique variable $x_{A,1}$ or $x_{A,5}$ on a position $(x',y')\in O$. Again this would imply that $M[x',y']\neq 0$ in contradiction to our assumption. The same holds for M_2, M_3 and M_4. So we can assume that $M_i[x,y]=0$ for $1\leq i\leq 4$ and all $(x,y)\in O$. By the induction assumption this means that $M_1=T_{k,j}^a$, $M_2=T_{i,k}^b$, $M_3=T_{k,j}^c$ and $M_4=T_{i,k}^d$ and that $-a,c\in[B]_\mathcal{C}$ and $b,-d\in[C]_\mathcal{C}$. By Lemma 9.5 we get

$$M=\left(T_{4,k}^{x_{A,9}}T_{k,4}^{-x_{A,10}}\right)T_{k,j}^aT_{i,k}^bT_{k,j}^cT_{i,k}^d\left(T_{k,4}^{x_{A,10}}T_{4,k}^{-x_{A,9}}\right).$$

Now assume $a+c\neq 0$. Then $(\prod_{i=1}^4 M_i)[k,j]=a+c\neq 0$ and

$$\left(\left(T_{4,k}^{x_{A,9}}T_{k,4}^{-x_{A,10}}\right)\prod_{i=1}^4 M_i\left(T_{k,4}^{x_{A,10}}T_{4,k}^{-x_{A,9}}\right)\right)[4,j]=M[4,j]=x_{A,9}(a+c)\neq 0,$$

which again contradicts our assumption. So we get $-a=c$. With the same argumentation we get $b=-d$ and with Lemma 9.5

$$\begin{aligned}M&=\left(T_{4,k}^{x_{A,9}}T_{k,4}^{-x_{A,10}}\right)T_{k,j}^{-c}T_{i,k}^bT_{k,j}^cT_{i,k}^{-b}\left(T_{k,4}^{x_{A,10}}T_{4,k}^{-x_{A,9}}\right)\\&=\left(T_{4,k}^{x_{A,9}}T_{k,4}^{-x_{A,10}}\right)T_{i,j}^{bc}\left(T_{k,4}^{x_{A,10}}T_{4,k}^{-x_{A,9}}\right)\\&=T_{i,j}^{bc}.\end{aligned}$$

Finally, since $b\in[B]_\mathcal{C}$ and $c\in[C]_\mathcal{C}$, it holds $bc\in[B]_\mathcal{C}\cdot[C]_\mathcal{C}=[A]_\mathcal{C}$.

□

So Lemma 9.6 holds and we get that $\mathsf{CIP}(\mathsf{SL}_5(\mathbb{Z}[x_1,\ldots,x_k]))$ is $\mathsf{NEXPTIME}$-hard. Now we show that we can reduce CIP for $\mathsf{SL}_5(\mathbb{Z}[x_1,\ldots,x_k])$ to CIP for $\mathsf{SL}_5(\mathbb{Z})$: Let $\mathcal{D}$ be a circuit over $\mathcal{P}(\mathsf{SL}_5(\mathbb{Z}[x_1,\ldots x_k]))$. By induction we see that for $n=|\mathcal{D}|$ the entries of the matrices in $[\mathcal{D}]$ are polynomials with the following properties:

- The number of monomials in every polynomial is bounded by $5^{2^{2n}}$.
- The coefficients are bounded by $5^{2^{2n}}$.
- The degrees are bounded by 2^n.

Since the proof of Proposition 2.1 in [6] essentially also works for these bounds, we know that for $B_{n,i} = 2^{2^{in^2}}$ for every of the polynomials $P(x_1, \ldots, x_k)$ that occurs in a matrix in $[\mathcal{D}]$ holds $P(x_1, ..., x_k) = 0$ exactly if $P(B_{n,1}, \ldots, B_{n,k}) = 0$.

Now we define a circuit $\mathcal{D}'$ over $\mathcal{P}(\mathsf{SL}_5(\mathbb{Z}))$ in the following way: starting with the circuit $\mathcal{D}$, we add gates $X_{i,j,d,m}$ for $1 \leq i, j, \leq 5$, $d \in \{-1, 1\}$ and $0 \leq m \leq kn^2$. Then we replace every input gate that evaluates to $T_{i,j}^{dx_l}$ with $1 \leq l \leq k$ by the gate X_{i,j,d,ln^2}. With some $h \in \{1, \ldots, 5\} \setminus \{i, j\}$ and for $1 \leq m \leq kn^2$ we set

$$\mathsf{rhs}'(X_{i,j,1,m}) = X_{h,j,-1,m-1}X_{i,h,1,m-1}X_{h,j,1,m-1}X_{i,h,-1,m-1},$$

$$\mathsf{rhs}'(X_{i,j,-1,m}) = X_{h,j,-1,m-1}X_{i,h,-1,m-1}X_{h,j,1,m-1}X_{i,h,1,m-1},$$

and $rhs'(X_{i,j,0}) = T_{i,j}^2$. Then $[X_{i,j,d,m}]_{\mathcal{D}'} = T_{i,j}^{d2^{2^m}}$, $[X_{i,j,d,ln^2}]_{\mathcal{D}'} = T_{i,j}^{B_{n,l}}$ and we get that $\mathsf{Id} \in [\mathcal{D}]$ exactly if $\mathsf{Id} \in [\mathcal{D}']$. This finishes the proof of Theorem 9.4.

Chapter 10

Overview and outlook

In Chapter 4 it was shown that the classical word problem for a f.g. linear group G is DLOGTIME-uniform TC^0-complete, if G is infinite solvable and in DLOGTIME-uniform NC^1 if G is virtually solvable (Theorem 4.2).

The following tables give a summarizing overview about the results for the circuit evaluation problem in this thesis and the results that were used.

If not marked differently we assume that in the following table k is part of the input.

Unitary rings	Additional properties	Complexity	Reference
$(\mathbb{Z}, +, \cdot)$	constant multiplicative depth	$\mathsf{C}_=\mathsf{L}$-complete	Lemma 5.10
$(\mathbb{Z}, +, \cdot)$	poly. bounded formal degree	$\mathsf{C}_=\mathsf{LogCFL}$-complete	Lemma 5.13
$(\mathbb{Z}, +, \cdot)$	-	coRP	[6]
$(\mathbb{Z}[x_1, \ldots, x_k], +, \cdot)$	skew, k constant	$\mathsf{C}_=\mathsf{L}$	Lemma 5.19
$(\mathbb{Z}[x_1, \ldots, x_k], +, \cdot)$	skew	coRNC	[48]
$(\mathbb{Z}[x_1, \ldots, x_k], +, \cdot)$	powerful skew	coRNC	Theorem 6.1
$(\mathbb{Z}[x_1, \ldots, x_k], +, \cdot)$	-	coRP	[45]
$(\mathbb{Z}_n[x_1, \ldots, x_k], +, \cdot)$	skew, k constant	DET	Lemma 5.19
$(\mathbb{Z}_n[x_1, \ldots, x_k], +, \cdot)$	skew	coRNC	[48]
$(\mathbb{Z}_p[x_1, \ldots, x_k], +, \cdot)$	powerful skew p prime	coRNC	Theorem 6.1
$(\mathbb{Z}_n[x_1, \ldots, x_k], +, \cdot)$	-	coRP	[45]

In the following table let

- G_1 be a direct product of finitely many copies of $\mathbb{Z}$ and $\mathbb{Z}_p$ for p prime,
- G_2 be f.g. abelian,
- H be f.g. virtually abelian,
- F_k be the free group of rank k with a normal subset N such that $F_k/[N, N]$ is f.g. virtually abelian, and
- $G_{1+\sqrt{2}}$ be the polycyclic group defined in section 7.6.

F.g. group	Additional properties	Complexity	Reference
$(\mathbb{Z}, +)$	-	$\mathsf{C}_=\mathsf{L}$-complete	[84]
nilpotent	torsion-free, non-trivial	$\mathsf{C}_=\mathsf{L}$-complete	Theorem 7.10
nilpotent	-	DET	Theorem 7.16
linear	–	coRP	[57]
$G_1 \wr H$	-	coRNC	Corollary 7.20
$G_2 \wr H$	-	coRP	Corollary 7.21
$G_2 \wr H$	H is finite	DET	Corollary 7.21
$F_k/[N, N]$	-	coRNC	Theorem 7.23
$G_{1+\sqrt{2}}$	-	$\geq_{\mathsf{L}}$ PIT($\mathbb{Z}$) for powerful skew circuits	Theorem 7.25

Finite semigroup	Additional properties	Complexity	Reference
non-solvable finite semigroup	-	P-complete	[19]
solvable finite semigroup	-	DET	[19]
acyclic semigroup	-	$\mathsf{AC}^0(\mathsf{NL})$	[19]

Finite semiring	Additional properties	Complexity	Reference
$\mathbb{B}_2$	-	P-complete	[52]
$(\mathbb{Z}_n, +, \cdot)$	-	P-complete	Lemma 8.1
R	$R^\bullet$ non-solvable	P-complete	[19]
R	R not $\{0,1\}$-free	P-complete	Theorem 8.2
R	$R^\bullet$ solvable, R $\{0,1\}$-free	DET	Theorem 8.2

The last table summarizes the results about the circuit intersection problem:

Structure	Additional properties	Complexity	Reference
finite semigroup S	S is a solvable local group	DET	Theorem 9.3
finite semigroup S	S contains a monoid	P-complete	Theorem 9.3
$(\mathbb{Z}, +)$	-	NP-complete	[62]
F_2	-	PSPACE-complete	[54]
$\mathsf{SL}_5(\mathbb{Z})$	-	NEXPTIME-complete	Theorem 9.4
$(\mathbb{Z}, +, \cdot)$	-	NEXPTIME-complete	[84]

Additionally it was shown that equality-testing for n-dimensional SLPs is in coRNC (Theorem 6.7).

There are still many open questions concerning circuit evaluation for various algebraic structures: in the context of groups it would be interesting to know a better upper complexity bound for circuit evaluation for polycyclic groups. Of course we know that the problem is in coRP, since polycyclic groups are linear, but there is some evidence that this problem could be in coRNC: since polycyclic groups are triangularizable over some ring R, the corresponding circuit looks similarly to the one for torsion-free nilpotent groups, but since the diagonal elements are not equal to 1, one can get large powers of elements. So these circuits can be seen as some kind of powerful circuits, where on the inputs there are elements with large powers. But since it is a famous open problem, whether modular powering is in NC, it is not clear even for $R = \mathbb{Z}$, whether those circuits can be evaluated in coRNC. On the other hand Nikolaev presented recently a joint work with Ushakov,

where it was shown that for polycyclic groups that are not virtually nilpotent the subset sum problem (for further information see e.g.[64]) is NP-complete. It seems possible that with their techniques one can show that for every polycyclic group that is not virtually nilpotent circuit evaluation is as least as hard as polynomial identity testing for powerful skew circuits over $\mathbb{Z}$.

In [70] the authors considered circuit evaluation for finite groupoids (a groupoid is a set S with an operation $\cdot$ that, in contrast to semigroups, need not to be associative). They found a subclass of groupoids (so-called polyabelian groupoids) where circuit evaluation is in DET. Two questions arise: on the one hand it could be possible to enlarge this subclass, since they give an example for a groupoid S that is not polyabelian, but one can see with the ideas from Chapter 8, that circuit evaluation for this groupoid is in DET. On the other hand, one could try to extend the result to some algebraic structure $(S, +, \cdot)$ where $(S, \cdot)$ is a groupoid and $(S, +)$ is a commutative semigroup as it was done in Chapter 8 for semigroups. Picking up on this idea one could also extend the setting from finite semirings to a structure where $+$ is not commutative anymore. The final goal in the setting of finite structures would be to show that for every finite algebraic structure, circuit evaluation is either in DET or P-complete, i.e., that there are no P-intermediate circuit evaluation problems for finite structures. This can be seen analogously to the famous conjecture of Feder and Vardi, that every constraint-satisfaction problem is either in P or NP-complete. See e.g. [88] for more information about this.

It would be also interesting to see, whether our techniques from Chapter 8 can be extended to get some results for the circuit intersection problem for infinite semigroups. As we have seen in Chapter 9 one should not hope for efficient algorithms in that setting. Here it would be interesting to determine the exact complexity of the circuit intersection problem for unitriangular groups. We know that this problem is in PSPACE and NP-hard, but even for the so-called discrete Heisenberg group $\mathsf{UT}_3(\mathbb{Z})$ no better bounds are known.

Our coRNC^2 identity testing algorithm for powerful skew circuits only works for the coefficient rings $\mathbb{Z}$ and $\mathbb{Z}_p$ with p prime. It is not clear how to extend it to $\mathbb{Z}_n$ with n composite. The Agrawal-Biswas identity testing algorithm also works for $\mathbb{Z}_n$ with n composite. But the problem is that the Fich-Tompa algorithm only works for polynomial rings over $\mathbb{Z}_p$ with p prime. For a f.g. abelian group G and a f.g. virtually abelian group H it remains open whether $\mathsf{CEP}(G \wr H)$ is in coRNC. In the context of polynomial identity testing Arvind, Mukhopadhyay and Raja showed recently [14] that circuits over non-commutative polynomials can be evaluated in coRP, if the number of monomials in the represented polynomials is bounded by 2^n. This could lead to results for circuit evaluation for wreath products over free groups.

For equality testing for multi-dimensional straight-line programs it remains open whether a polynomial time algorithm exists. For the one-dimensional (string) case, a polynomial time algorithm exists. Here, it remains open, whether equality testing is in NC.

Publications

[1] D. König and M. Lohrey. Evaluating matrix circuits. *Proceedings of the 21st International Conference on Computing and Combinatorics, COCOON 2015*, volume 9198 of *Lecture Notes in Computer Science*, 235–248. *Springer*, 2015.

[2] D. König and M. Lohrey. Parallel identity testing for skew circuits with big powers and applications. *Proceedings of the 40th International Symposium on Mathematical Foundations of Computer Science 2015, MFCS 2015, Part II*, volume 9235 of *Lecture Notes in Computer Science*, 445–458. *Springer*, 2015.

[3] M. Ganardi, D.Hucke, D. König and M. Lohrey. Circuit evaluation for finite semirings. *Proceedings of the 34th International Symposium on Theoretical Aspects of Computer Science, STACS 2017*, volume 66 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 35:1–35:14, 2017.

D. König, M. Lohrey and G. Zetsche. Knapsack and subset sum problems in nilpotent, polycyclic, and co-context-free groups. *Contemporary Mathematics 677 (Algebra and Computer Science)*, 129–144, 2016.

Bibliography

[4] M. Agrawal and S. Biswas. Primality and identity testing via chinese remaindering. *Journal of the Association for Computing Machinery*, 50(4):429–443, 2003.

[5] E. Allender, R. Beals, and M. Ogihara. The complexity of matrix rank and feasible systems of linear equations. *Computational Complexity*, 8(2):99–126, 1999.

[6] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. B. Miltersen. On the complexity of numerical analysis. *SIAM Journal on Computing*, 38(5):1987–2006, 2009.

[7] E. Allender, J. Jiao, M. Mahajan, and V. Vinay. Non-commutative arithmetic circuits: Depth reduction and size lower bounds. *Theoretical Computer Science*, 209(1-2):47–86, 1998.

[8] J. Almeida, S. Margolis, B. Steinberg, and M. Volkov. Representation theory of finite semigroups, semigroup radicals and formal language theory. *Transactions of the American Mathematical Society*, 361(3):1429–1461, 2009.

[9] C. Àlvarez, J. L. Balcázar, and B. Jenner. Functional oracle queries as a measure of parallel time. In *Proceedings of the 8th Annual Symposium on Theoretical Aspects of Computer Science, STACS 1991*, volume 480 of *Lecture Notes in Computer Science*, 422–433. *Springer*, 1991.

[10] C. Àlvarez and B. Jenner. A very hard log-space counting class. *Theoretical Computer Science*, 107(1):3–30, 1993.

[11] S. Arora and B. Barak. Computational complexity - a modern approach. *Cambridge University Press*, 2009.

[12] E. Artin. Theorie der Zöpfe. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 4(1):47–72, 1925. In German.

[13] V. Arvind and P. S. Joglekar. Arithmetic circuit size, identity testing, and finite automata. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:26, 2009.

[14] V. Arvind, P. Mukhopadhyay and S. Raja Randomized polynomial time identity testing for noncommutative circuits. arXiv:1606.00596, 2016.

[15] K. Auinger and B. Steinberg. Constructing divisions into power groups. *Theoretical Computer Science*, 341(1–3):1–21, 2005.

[16] L. Auslander. On a problem of Philip Hall. *Annals of Mathematics*, 86(2):112–116, 1967.

[17] D. A. M. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences*, 38:150–164, 1989.

[18] D. A. M. Barrington and D. Thérien. Finite monoids and the fine structure of NC^1. *Journal of the Association for Computing Machinery*, 35(4):941–952, 1988.

[19] M. Beaudry, P. McKenzie, P. Péladeau, and D. Thérien. Finite monoids: From word to circuit evaluation. *SIAM Journal on Computing*, 26(1):138–152, 1997.

[20] M. Ben-Or and R. Cleve. Computing algebraic formulas using a constant number of registers. *SIAM Journal on Computing*, 21(1):54–58, 1992.

[21] P. Berman, M. Karpinski, L. L. Larmore, W. Plandowski, and W. Rytter. On the complexity of pattern matching for highly compressed two-dimensional texts. *Journal of Computer and System Sciences*, 65(2):332–350, 2002.

[22] D. K. Biss and S. Dasgupta. A presentation for the unipotent group over rings with identity. *Journal of Algebra*, 237(2):691–707, 2001.

[23] W. W. Boone. The word problem. *Annals of Mathematics. Second Series*, 70:207–265, 1959.

[24] F. B. Cannonito. Hierachies of computable groups and the word problem. *Journal of Symbolic Logic*, 32:376–392, 1966.

[25] F. B. Cannonito and R. W. Gatterdam. The word problem in polycyclic groups is elementary. *Composito Mathematica*, 27: 39–45, 1973.

[26] F. B. Cannonito and R. W. Gatterdam. The word problem and power problem in 1-relator groups are primitive recursive. *Pacific Journal of Mathematics*, 61(2):351–359, 1975.

[27] A. Chiu, G. Davida, and B. Litow. Division in logspace-uniform NC^1. *Theoretical Informatics and Applications. Informatique Théorique et Applications*, 35(3):259–275, 2001.

[28] S. A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64:2–22, 1985.

[29] S. A. Cook and P. Nguyen. Logical Foundation of Proof Complexity. *Cambridge University Press*, 2010.

[30] S. A. Cook and L. Fontes. Formal theories for linear algebra. *Logical Methods in Computer Science*, 8(1), 2012.

[31] M. Dehn. Über unendliche diskontinuierliche Gruppen. *Mathematische Annalen*, 71:116–144, 1911. In German.

[32] M. Dehn. Transformation der Kurven auf zweiseitige Flächen. *Mathematische Annalen*, 72:413–421, 1912. In German.

[33] R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4): 193–195, 1978.

[34] V. Diekert, A. G. Myasnikov, and A. Weiß. Conjugacy in Baumslag's group, generic case complexity, and division in power circuits. In *Proceedings of the 11th Symposium on Latin American Theoretical Informatics, LATIN 2014*, volume 8392 of *Lecture Notes in Computer Science*, 1–12. *Springer*, 2014.

[35] W. Eberly. Very fast parallel polynomial arithmetic. *SIAM Journal on Computing*, 18(5):955–976, 1989.

[36] S. Eilenberg, and M. P. Schützenberger. On pseudovarieties. *Advances in Mathematics, Vol. 19, No.3, Academic Press*, New York and London, 1976.

[37] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P. Thurston. Word Processing in Groups. *Jones and Bartlett*, Boston, 1992.

[38] F. E. Fich and M. Tompa. The parallel complexity of exponentiating polynomials over finite fields. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pages 38–47. ACM, 1985.

[39] J. S. Golan. Semirings and their Applications. *Springer*, 1999.

[40] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. Limits to Parallel Computation: *P*-Completeness Theory. *Oxford University Press*, 1995.

[41] M. Gromov. Hyperbolic groups. In S. M. Gersten, editor, *Essays in Group Theory*, number 8 in MSRI Publ., 75–263. *Springer*, 1987.

[42] O. G. Harlampovič. A finitely presented solvable group with unsolvable word problem.' (Russian) *Izv. Akad. Nauk SSSR Ser. Mat. 45* no. 4, 852–873, 928, 1981.

[43] W. Hesse, E. Allender, and D. A. M. Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *Journal of Computer and System Sciences*, 65:695–716, 2002.

[44] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. *Theoretical Computer Science*, 158(1&2):143–159, 1996.

[45] O. H. Ibarra and S. Moran. Probabilistic algorithms for deciding equivalence of straight-line programs. *Journal of the Association for Computing Machinery*, 30(1):217–228, 1983.

[46] R. Impagliazzo and A. Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing, STOC 1997*, 220–229. *ACM Press*, 1997.

[47] A. Jeż. Faster fully compressed pattern matching by recompression. *Proceedings of the 39th International Colloquium on Automata, Languages and Programming, ICALP 2012*, volume 7391 of *Lecture Notes in Computer Science*, 533 –544. *Springer*, 2012.

[48] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

[49] I. Kapovich, A. Miasnikov, P. Schupp, and V. Shpilrain. Generic-case complexity, decision problems in group theory, and random walks. *Journal of Algebra*, 264(2):665–694, 2003.

[50] M. I. Kargapolov and J. I. Merzljakov. Fundamentals of the Theory of Groups, volume 62 of *Graduate Texts in Mathematics. Springer*, New York, 1979.

[51] S. R. Kosaraju. On parallel evaluation of classes of circuits. In *Proceedings of the 10th Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 1990*, volume 472 of *Lecture Notes in Computer Science*, 232–237. *Springer*, 1990.

[52] R. E. Ladner. The circuit value problem is log space complete for P. *SIGACT News*, 7(1):18–20, 1975.

[53] R. J. Lipton and Y. Zalcstein. Word problems solvable in logspace. *Journal of the Association for Computing Machinery*, 24(3):522–526, 1977.

[54] M. Lohrey. Word problems and membership problems on compressed words. *SIAM Journal on Computing*, 35(5):1210 – 1240, 2006.

[55] M. Lohrey. Algorithmics on SLP-compressed strings: A survey. *Groups Complexity Cryptology*, 4(2):241–299, 2012.

[56] M. Lohrey, B. Steinberg, and G. Zetzsche. Rational subsets and submonoids of wreath products. *Information and Computation*, 2014.

[57] M. Lohrey. The Compressed Word Problem for Groups. *SpringerBriefs in Mathematics. Springer*, 2014.

[58] M. Lohrey. Rational subsets of unitriangluar groups. *International Journal of Algebra and Computation*, 25(1-2):113–121, 2015.

[59] W. Magnus. Das Identitätsproblem für Gruppen mit einer definierenden Relation. *Mathematische Annalen*, 106(1):295–307, 1932. In German.

[60] W. Magnus. On a theorem of Marshall Hall. *Annals of Mathematics. Second Series*, 40:764–768, 1939.

[61] D. J. McCarthy and D. L. Hayes. Subgroups of the power semigroup of a group. *Journal of Combinatorial Theory, Series A*, 14(2):173–186, 1973.

[62] P. McKenzie and K. W. Wagner. The complexity of membership problems for circuits over sets of natural numbers. *Computational Complexity*, 16(3):211–244, 2007.

[63] K. Mehlhorn, R. Sundar, and C. Uhrig. Maintaining dynamic sequences under equality tests in polylogarithmic time. *Algorithmica*, 17(2):183–198, 1997.

[64] A. Miasnikov, A. Nikolaev and A. Ushakov. Knapsack problems in groups. *Mathematics of Computation*, 84:987–1016, 2015.

[65] A, Miasnikov, S. Vassileva and A. Weiß. The conjugacy problem in free solvable groups and wreath products of abelian groups is in TC^0. To appear in *Proceedings of CSR 2017*.

[66] A, Miasnikov and A. Weiß. TC^0 circuits for algorithmic problems in nilpotent groups. arXiv:1702.06616, 2017.

[67] G. Miller. The commutator subgroup of a group generated by two operators. *Proceedings of the National Academy of Sciences of the United States of America*, 18:665–668, 1932.

[68] G. L. Miller, V. Ramachandran, and E. Kaltofen. Efficient parallel evaluation of straight-line code and arithmetic circuits. *SIAM J. Comput.*, 17(4):687–695, 1988.

[69] G. L. Miller and S. Teng. The dynamic parallel complexity of computational circuits. *SIAM J. Comput.*, 28(5):1664–1688, 1999.

[70] C. Moore, D. Thérien, F. Lemieux, J. Berman, and A. Drisko. Circuits and expressions with nonassociative gates. *J. Comput. Syst. Sci.*, 60(2):368–394, 2000.

[71] P. S. Novikov. On the algorithmic unsolvability of the word problem in group theory. *American Mathematical Society, Translation, II. Series*, 9: 1–122, 1958.

[72] W. Plandowski. Testing equivalence of morphisms on context-free languages. In *Proceedings of the 2nd Annual European Symposium on Algorithms, ESA 1994*, volume 855 of *Lecture Notes in Computer Science*, pages 460–470. *Springer*, 1994.

[73] M. Rabin. Computable algebra, general theory and theory of computable fields. *Transactions of the American Mathematical Society*, 95:341–360, 1960.

[74] J. Rhodes and B. Steinberg. The q-theory of Finite Semigroups. *Springer*, 2008.

[75] D. Robinson. Parallel Algorithms for Group Word Problems. *PhD thesis, University of California, San Diego*, 1993.

[76] J. J. Rotman. An Introduction to the Theory of Groups (fourth edition). *Springer*, 1995.

[77] S. Schleimer. Polynomial-time word problems. *Commentarii Mathematici Helvetici*, 83(4):741–765, 2008.

[78] J. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27:701–717, 1980.

[79] H.-U. Simon. Word problems for groups and contextfree recognition. In *Proceedings of Fundamentals of Computation Theory, FCT 1979*, 417–422. *Akademie-Verlag*, 1979.

[80] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time (preliminary report). In *Proceedings of the 5th Annual ACM Symposium on Theory of Computing (STOCS 73)*, 1–9. ACM Press, 1973.

[81] R. Swan. Representations of polycyclic groups. *Proceedings of the American Mathematical Society*, 18:573–574, 1967.

[82] J. Tits. Free subgroups in linear groups. *Journal of Algebra*, 20:250–270, 1972.

[83] S. Toda. Counting problems computationally equivalent to computing the determinant. *Technical Report CSIM 91-07, University of Electro-Communications, Tokyo*, 1991.

[84] S. D. Travers. The complexity of membership problems for circuits over sets of integers. *Theor. Comput. Sci.*, 369(1-3):211–229, 2006.

[85] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983.

[86] V. Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, 270–284. IEEE Computer Society, 1991.

[87] H. Vollmer. Introduction to Circuit Complexity. *Springer*, 1999.

[88] H. Vollmer. The gap-language technique revisited. In *4th Computer Science Logic, Selected Papers,* volume 533 of *Lecture Notes in Computer Science*, 389–399, *Springer*, 1991.

[89] S. Waack. On the parallel complexity of linear groups. *R.A.I.R.O. — Informatique Théorique et Applications*, 25(4):265–281, 1991.

[90] B. A. F. Wehrfritz. Infinite Linear Groups. *Springer*, 1977.

[91] A. Weiß. On the Complexity of Conjugacy in Amalgamated Products and HNN Extensions. *PhD thesis, Universität Stuttgart*, 2015.

[92] O. Zariski and P. Samuel. Commutative Algebra, Volume I, volume 28 of *Graduate Texts in Mathematics. Springer*, 1958.

[93] R. Zippel. Probabilistic algorithms for sparse polynomials. In *Ng E.W. (eds) Symbolic and Algebraic Computations*, volume 72 of *Lecture Notes in Computer Science*, 216–226. *Springer*, 1979.