Algorithms and Complexity Results for Finite Semigroups

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Lukas Fleischer

aus Bietigheim-Bissingen

Hauptberichter: Prof. Dr. Volker Diekert

Mitberichter: Prof. Dr. Pascal Weil

Tag der mündlichen Prüfung: 20. Februar 2019

Institut für Formale Methoden der Informatik
Universität Stuttgart

2019

Contents

Abstract 5

Zusammenfassung 7

1 Introduction 9

2 Preliminaries 15

2.1 Formal Languages . . . 15

2.2 Algebra . . . 15

2.3 Recognizable Languages and Descriptional Complexity . . . 28

2.4 Complexity Theory . . . 29

3 Building Blocks for Efficient Algorithms 37

3.1 First-Order Definable Properties . . . 37

3.2 Cayley Circuits and Straight-Line Programs . . . 42

3.3 Computing Powers, Indices and Periods . . . 47

4 The Circuits Properties Framework 49

4.1 Circuits Properties . . . 51

4.2 Product Circuits Properties . . . 57

5 Decision Problems 63

5.1 Subsemigroup Membership . . . 63

5.2 The Word Problem . . . 71

5.3 Language Properties . . . 71

5.4 Intersection Non-Emptiness . . . 78

5.5 Variety Membership . . . 87

6 Conclusion and Future Work 93

Abstract

We consider the complexity of decision problems for regular languages given as recognizing morphisms to finite semigroups. We describe efficient algorithms for testing language *emptiness*, *universality*, *inclusion*, *equivalence* and *finiteness*, as well as *intersection non-emptiness*. Some of these algorithms have sublinear running time and are therefore implemented on random-access Turing machines or Boolean circuits. These algorithms are complemented by lower bounds. We give completeness results for the general case and also investigate restrictions to certain varieties of finite semigroups.

Except for intersection non-emptiness, the problems mentioned above are shown to be closely connected to the *Cayley semigroup membership problem*, i.e., membership of an element to a subsemigroup given by a multiplication table and a set of generators. Therefore, the complexity of this problem is one of the main topics of this thesis. In many (but not all) cases, efficient algorithms for Cayley semigroup membership are based on the existence of succinct representations of semigroup elements over a given set of generators. These representations are algebraic circuits, also referred to as *straight-line programs.* As a compressibility measure for such representations within specific classes of finite semigroups, we introduce a framework called *circuits properties.* We give algebraic characterizations of certain classes of circuits properties and derive complexity results. As a byproduct, a generalization of a long-standing open problem in complexity theory is resolved. For intersection non-emptiness, a similar tool called *product circuits properties* is used.

We provide completeness results for the problem of deciding membership to varieties of finite semigroups and to varieties of languages. We show that many varieties, which were previously known to be decidable in polynomial time, are actually in $\mathsf{DLOGTIME}$-uniform AC^0. The key ingredient is definability of varieties by first-order formulas. Combining our results with known lower bounds for deciding Parity, we also present a novel technique to prove that a specific variety cannot be defined by first-order formulas with multiplication. Since such formulas are more expressive than finite sets of ω-identities, this implies non-definability by finite sets of ω-identities.

Zusammenfassung

Diese Arbeit befasst sich mit der Komplexität von Entscheidungsproblemen für reguläre Sprachen, wobei diese als erkennende Homomorphismen zwischen freien und endlichen Halbgruppen repräsentiert werden. Es werden effiziente Algorithmen für das Testen einer Sprache auf *Leerheit*, *Universalität*, *Inklusion*, *Äquivalenz* und *Endlichkeit* sowie für das Testen auf *Leerheit des Schnitts von Sprachen* beschrieben. Einige dieser Algorithmen haben sublineare Laufzeit und werden daher auf Turingmaschinen mit Direktzugriff auf die Eingabe oder auf booleschen Schaltkreisen implementiert. Diese Algorithmen werden durch entsprechende untere Schranken ergänzt. Für einige dieser Probleme, sowie deren Einschränkung auf bestimmte Varietäten endlicher Halbgruppen, wird Vollständigkeit für bekannte Komplexitätsklassen nachgewiesen.

Ausgenommen der Leerheit des Schnitts, besteht eine enge Verwandtschaft der zuvor genannten Probleme mit dem *Cayley Semigroup Membership Problem*, d.h. Zugehörigkeit eines Elementes zu einer durch Multiplikationstabelle und Erzeuger gegebenen Unterhalbgruppe. Daher ist die Komplexität dieses Problems einer der Schwerpunkte dieser Arbeit. In vielen (jedoch nicht allen) Fällen basieren effiziente Algorithmen für die Zugehörigkeit zu Unterhalbgruppen auf der Existenz kompakter Darstellungen von Elementen über einer gegebenen Erzeugendenmenge. Für diese kompakte Darstellung werden algebraische Schaltkreise, auch *Straight-Line-Programme* genannt, verwendet. Als Maß für die Komprimierbarkeit solcher Darstellungen innerhalb bestimmter Klassen endlicher Halbgruppen werden sogenannte *Schaltkreis-Eigenschaften* eingeführt. Es werden algebraische Charakterisierungen für bestimmte Klassen von Schaltkreis-Eigenschaften gegeben und Komplexitätsresultate abgeleitet. Als Nebenprodukt wird eine Verallgemeinerung eines seit langem offenen Problems aus der Komplexitätstheorie gelöst. Für die Leerheit des Schnitts wird ein ähnliches Werkzeug, genannt *Produkt-Schaltkreis-Eigenschaften*, verwendet.

Es werden Komplexitätsresultate für das Entscheidungsproblem der Zugehörigkeit einer endlichen Halbgruppe oder einer Sprache zu einer Varietät vorgestellt. Für viele Varietäten, für die bereits bekannt war, dass der Zugehörigkeitstest in Polynomialzeit entscheidbar ist, wird gezeigt, dass dieser Test sogar in der $\mathsf{DLOGTIME}$-uniformen Variante der Schaltkreis-Komplexitätsklasse AC^0 liegt. Die Kernkomponente dieses Resultats ist die Definierbarkeit von Varietäten durch prädikatenlogische Formeln erster Stufe. Eine Kombination der Resultate mit bekannten unteren Schranken für die Paritätsfunktion liefert eine neue Technik, um nachzuweisen, dass eine bestimmte Varietät nicht durch Formeln erster Stufe mit Multiplikation definierbar ist. Da solche Formeln ausdrucksstärker als endliche Mengen von ω-Termen sind, folgt aus einem solchen Nachweis auch Nicht-Definierbarkeit durch endlich viele ω-Terme.

Chapter 1

Introduction

The class of regular languages is well-understood in formal language theory. It is the centerpiece of a well-founded theory with connections to many different fields of theoretical computer science. Regular languages have numerous equivalent characterizations in terms of *rational expressions*, *regular grammars*, *finite automata*, *finite semigroups* and *monadic second-order logic over words*. Each of these characterizations has its own merits and limitations. Rational expressions are a concise and powerful tool for describing patterns in strings. They are widely used for *pattern matching* and can be found in software such as search engines, word processors, text processing utilities and compilers. Logics over words provide a simple yet expressive way to describe properties of languages which makes them a useful tool in *model checking*, a subdiscipline of *software verification*. Finite automata, in particular the *deterministic* variant, are a compact representation of regular languages, with efficient algorithms for commonly occurring operations (such as Boolean operations or concatenation of languages). They also admit efficient procedures for common decision problems (such as language emptiness or inclusion). Hence, they are often the internal data structure of choice to represent and process regular languages in implementations. Finite semigroups have proven to be a very convenient concept when it comes to investigating decidability questions and structural properties of regular languages. Also, recent work by Kufleitner and the author [FK15] suggests that finite semigroups might be a viable alternative to automata as data structures in the context of *infinite words*: here, the commonly used automaton models are much less efficient than over finite words; a trade-off between size and the existence of efficient algorithms makes finite semigroups more attractive than in the case of finite words. Regular languages over infinite words (sometimes also referred to as *ω-regular languages*) play an important role in model checking.

Kleene's theorem [Kle56] on the equivalence of regular expressions and finite automata is often considered the starting point of automata theory. Further results followed shortly afterwards, with increasing interest in considering subclasses of the class of regular languages. One of the earliest and most famous results of this type is Schützenberger's theorem on *star-free languages*: replacing the Kleene star operator in rational expressions by complementation yields a strict subset of the regular languages that corresponds to the class of *aperiodic* semigroups [Sch65]. One can also characterize this class by *counter-free automata* and in logics, this class is equivalent to languages definable by *first-order* formulas. Several years later, with the rise of complexity theory in computer science, work on the computational complexity of several decision problems on representations of regular languages followed. Stockmeyer and Meyer's seminal work on decidable word problems from automata theory [MS72] is one of the most notable early contributions. Different people then started combining the study of subclasses of the regular languages with complexity investigations. This interdisciplinary work is motivated by the idea that such subclasses often correspond to natural restrictions on automata or semigroups (as illustrated in the case of star-free languages above), and considering restricted inputs often gives rise to more efficient algorithms.

While many interesting and insightful results were made in this area [Bar89, BT88, BCST92, BMT92], several problems turned out to be quite challenging and remained unresolved for several decades. A common theme in most existing work is that finite automata are usually chosen as input encoding, while restrictions are often formulated in algebraic terms. Our motivation for studying the (fully) algebraic variants of these problems with the inputs given as finite semigroups is threefold. Firstly, while finite automata are inarguably the classical and canonical model for representing regular languages, having algebraic objects seems like a natural choice in the context of algebraic constraints. Secondly, there is a simple and very weak form of reductions from semigroups to automata, so any lower bounds and hardness results obtained for semigroups immediately transfer to the automaton setting. On the other hand, certain problems might be — and some actually are — easier for semigroups than for automata; a circumstance that we expect to lead to new ideas and techniques which may ultimately stimulate progress in the automaton setting as well. The aforementioned observation that finite semigroups might have practical applications as data structures is the third driving factor.

Outline and Summary of Results. In Chapter 2, we provide the notation, notions and algebraic foundations required in this thesis as well as basics from complexity theory. We introduce the *polylogarithmic time hierarchy* which is captured by alternating random-access Turing machines with bounded alternation and polylogarithmic running time. We show that this hierarchy is contained in the circuit complexity class qAC^0 (unbounded fan-in Boolean circuits of quasi-polynomial size and constant depth). We give a very brief summary of the descriptional complexity aspects, pointing out the main implications of considering finite semigroups instead of finite automata as inputs.

In Chapter 3, we introduce important components for building and describing efficient algorithms on finite semigroups. The first tool is first-order logic over finite semigroups. We give examples of varieties of finite semigroups definable by first-order formulas and prove several closure properties. The second concept are *Cayley circuits* which are closely related to straight-line programs. We show that various decision problems on these objects can be solved efficiently by using non-deterministic random-access Turing machines. We also describe algorithms for efficient computation of powers, indices and periods of elements.

In Chapter 4, the notions of *circuits properties* and *products circuits properties* are presented. The special case of the *polylogarithmic circuits property* was first introduced by the author in [Fle18c] and the generalization to arbitrary functions and product circuits first appeared in [Fle18b]. This concept is fundamental to understanding efficient decision procedures for finite semigroups and will be heavily relied on in the subsequent chapter. In addition to previously published material, we give an algebraic characterization of the variety of finite monoids with the polylogarithmic circuits property. This property plays a particularly important role in the study of the *emptiness*, *universality*, *equivalence*, *inclusion* and *finiteness* problems and allows us to prove a dichotomy result for the complexity of these problems. The characterization is in terms of ω-identities and thus, both effective and efficiently decidable by results from Chapter 5. We also present additional previously unpublished fundamental results on circuits properties.

In Chapter 5, we finally study the complexity of various decision problems. The first section is devoted to studying the *Cayley membership problem*, a problem that has been investigated in different contexts before and turns out to be the key problem to solve for many language decision problems (this connection to other problems is established later in Section 5.3). It has been long known that the Cayley semigroup membership problem is NL-complete [JLL76]. However, for groups, Barrington and McKenzie observed back in 1991 that the problem can be reduced to reachability in undirected graphs and conjectured it to be L-complete [BM91]. This conjecture withstood resolution for over 25 years (see [BKLM01] for a partial result) and was refuted only recently by the author [Fle18c] using the circuits property framework to prove membership to the circuit complexity class qAC^0.

In this thesis, we further improve this result in two directions: we first show that it also holds for all *Clifford semigroups*, a proper superclass of the class of finite groups, as well as for commutative semigroups. Furthermore, we show that the problem belongs to $\mathsf{NPOLYLOGTIME}$, the first level of the polylogarithmic time hierarchy inside qAC^0. This complexity class is defined using polylogarithmic-time random-access Turing machines and we claim that it is actually a more natural model for solving the Cayley semigroup membership problem and related problems. Our claim is substantiated by two observations. Firstly, apart from non-determinism, previous algorithms to solve this kind of problem were sequential in nature, so a sequential model seems to be appropriate. Secondly, it allows us to establish a dichotomy theorem which reveals a close link between $\mathsf{NPOLYLOGTIME}$ and the polylogarithmic circuits property: for a variety of finite monoids $\mathbf{V}$, the problem is in $\mathsf{NPOLYLOGTIME}$ if and only if $\mathbf{V}$ has the polylogarithmic circuits property. Having the polylogarithmic circuits property is, in turn, shown to be equivalent to containing either only Clifford monoids or only commutative monoids. This dichotomy theorem is exceptional in different ways. Firstly, while it provides a fully algebraic characterization of monoid varieties with efficiently decidable Cayley semigroup membership, the boundary itself is not a variety. Indeed, the main techniques for proving that Clifford semigroups have the polylogarithmic circuits property are radically different from the techniques used in the commutative case. Secondly, the complexity part of the result is a “true” dichotomy statement in the sense that we can prove that Cayley semigroup membership is not in $\mathsf{NPOLYLOGTIME}$ for non-commutative non-Clifford monoids. The proof is unconditional and does not rely on any unproven computational hardness assumptions, as is often the case in complexity theory. This is made possible by the fact that polylogarithmic-time machines are a computationally weak model which makes proving lower bounds easy. We also study the Cayley semigroup membership problem for finite *bands*, i.e., classes of idempotent semigroups, and present some partial results for that case.

We briefly study the word problem for semigroups in Section 5.2. It is well-known that this problem belongs to L and is NC^1-hard. The *unary* word problem is shown to be decidable in $\mathcal{O}(\log^2 n)$ time on a deterministic random-access Turing machine. In contrast, for deterministic finite automata, both the unary and the unrestricted uniform word problem are L-complete.

Section 5.3 allows us to apply the complexity results for Cayley semigroup membership to other decision problems, such as the *emptiness*, *universality*, *equivalence* and *inclusion* problems for regular languages given as recognizing morphisms to finite semigroups. The *finiteness problem* for languages is studied as well. Moreover, we briefly describe how to transfer upper and lower bounds to the setting of infinite words.

We then investigate variants of the *intersection non-emptiness* problem. Here, many of the complexity results are based on the product circuits properties framework from Chapter 4. Some of the results from [FK18c] and [Fle18b] are presented in a more coherent way, and we provide slightly stronger statements as well as simplified proofs.

Lastly, we also consider the problem of deciding whether a given language or a given finite semigroup belongs to a certain *variety* and describe efficient decision procedures based on the logical formalism introduced in Chapter 3.

Our work also brings up several new open problems. We summarize our results and suggest further research directions in Chapter 6.

Previously Published Material. Parts of this thesis have been published in the following two conference papers:

- Lukas Fleischer. The Intersection Problem for Finite Semigroups. In *DLT 2018, Proceedings*, volume 11088 of *LNCS*, pages 318–329. Springer, 2018.
- Lukas Fleischer. On the Complexity of the Cayley Semigroup Membership Problem. In *CCC 2018, Proceedings*, volume 102 of *LIPIcs*, pages 25:1–25:12. Dagstuhl Publishing, 2018.

The results from Section 3.1 and in Section 5.5 have not been peer-reviewed before but have been made available as a technical report by the author [Fle18a]. The results from Section 3.3 are new and have not been published. Many results in Section 4.1 are novel, particularly the extension of the Babai-Szemerédi Reachability Lemma to Clifford semigroups and the dichotomy result for the polylogarithmic circuits property of varieties of finite monoids. Some results from Section 4.2, such as Lemma 4.21, have not been published previously. Section 5.1.1 extends previous results and in Section 5.1.2, previous hardness results are strengthened to obtain matching lower bounds for varieties of finite monoids. Most of the results on Cayley semigroup membership for *bands* presented in Section 5.1.3 are new as well (apart from those explicitly marked as having been known previously).

We also occasionally refer to the following article and recommend it for more details on the *intersection non-emptiness problem*. Some of the results are extended in this thesis.

- Lukas Fleischer and Manfred Kufleitner. The Intersection Problem for Finite Monoids. In *STACS 2018, Proceedings*, pages 30:1–30:14. Dagstuhl Publishing, 2018.

The following journal article investigates the complexity of variants of some of the decision problems considered in this work in the setting of infinite words. In this thesis, we only sketch how to adapt our results to this setting.

- Lukas Fleischer and Manfred Kufleitner. The complexity of weakly recognizing morphisms. *RAIRO-Theor. Inf. Appl.*, 2018.

Descriptional complexity aspects of finite semigroups and efficient string algorithms are only discussed briefly in the preliminaries of this thesis. The following conference and journal articles are recommended as complementary material for in-depth coverage of these topics.

- Lukas Fleischer and Manfred Kufleitner. Testing Simon's congruence. In *MFCS 2018, Proceedings*, volume 117 of *LIPIcs*, pages 62:1–62:13. Dagstuhl Publishing, 2018.

- Lukas Fleischer and Manfred Kufleitner. Green's Relations in Deterministic Finite Automata. *Theory of Computing Systems*, 2018.

- Lukas Fleischer and Manfred Kufleitner. Green's Relations in Finite Transformation Semigroups. In *CSR 2017, Proceedings*, volume 10304 of *LNCS*, pages 112–125. Springer, 2017.

- Lukas Fleischer and Manfred Kufleitner. Operations on Weakly Recognizing Morphisms. In *DCFS 2016, Proceedings*, volume 9777 of *LNCS*, pages 126–137. Springer, 2016.

The starting point of the research project behind this thesis was set with the following work based on ideas by Manfred Kufleitner and on results from the author's Master's thesis:

- Lukas Fleischer and Manfred Kufleitner. Efficient Algorithms for Morphisms over Omega-Regular Languages. In *FSTTCS 2015, Proceedings*, volume 45 of *LIPIcs*, pages 112–124. Dagstuhl Publishing, 2015.

Chapter 2

Preliminaries

In this chapter, we introduce the notation and notions used in this thesis. We state basic results from automata theory, algebra and complexity theory. Large parts of the presented material are not original; most results can be found in standard textbooks or are folklore. We try to give additional references when appropriate. Familiarity with fundamental concepts in mathematics and computer science, such as set theory, Big O notation and basic knowledge on the asymptotic growth of functions, is assumed. We will use $\mathbb{N}$ to denote the set of non-negative integers $\{0, 1, 2, \ldots\}$ and log to denote the binary logarithm.

2.1 Formal Languages

An *alphabet* is a non-empty set. Its elements are called *letters*. A *(finite) word* over an alphabet A is a finite sequence $w = a_1 \cdots a_n$ of letters $a_1, \ldots, a_n \in A$. The integer n is the *length* of the word w. It is denoted by $|w|$. The *empty word* of length 0 is denoted by ε. The *concatenation* uv of a word $u = a_1 \cdots a_\ell$ and a word $v = b_1 \cdots b_k$ is the word $a_1 \cdots a_\ell b_1 \cdots b_k$.

The *alphabet* (or *content*) of a word $w = a_1 \cdots a_\ell$ is the set $\{a_1, \ldots, a_\ell\}$ of all letters appearing in w. It is denoted by $\mathrm{alph}(w)$. For a letter $a \in A$, we use the notation $|w|_a$ to denote the number of occurrences of a in w, i.e., the number of positions $i \in \{1, \ldots, n\}$ such that $a_i = a$. For $n \in \mathbb{N}$, the set of all words of length n over A is denoted by A^n. Moreover, we let

$$A^{\leqslant n} = \bigcup_{i \leqslant n} A^i, \quad A^{\geqslant n} = \bigcup_{i \geqslant n} A^i, \quad A^+ = \bigcup_{i \geqslant 1} A^i, \quad \text{and} \quad A^* = \bigcup_{i \in \mathbb{N}} A^i.$$

2.2 Algebra

2.2.1 Semigroups and Homomorphisms

The main algebraic concepts used in this thesis are semigroups and monoids. A *semigroup* is a non-empty set equipped with an associative binary operation which is often also referred to as *multiplication*. We usually denote multiplication of two elements x, y by juxtaposition xy and sometimes use the notation $x \cdot y$ for clarification. A semigroup

M with a *neutral element*, i.e., an element $e \in M$ such that $ex = x = xe$ for all $x \in M$, is called *monoid*. The neutral element is unique and usually denoted by 1. For a semigroup S, we denote by S^1 the monoid obtained by adding a new neutral element to S. The notation S^{op} is used to denote the *opposite semigroup* of S, i.e., the semigroup obtained by replacing the binary operation on S by a new operation $x \circ y = yx$.

For an alphabet A, the set of all finite words A^* (resp. all non-empty finite words A^+) forms a monoid (resp. semigroup) with concatenation as multiplication. It is called the *free monoid* (resp. *free semigroup*) over A. Apart from free semigroups and free monoids, most semigroups considered in this work are finite.

An element x of a semigroup S is *idempotent* if $x^2 = x$ and the set of all idempotent elements of S is denoted by $E(S)$. In a finite semigroup S, for each element $x \in S$, there exist natural numbers $i, p > 0$ such that $x^{i+p} = x^i$. The smallest number i satisfying this equation is called *index* of the element x and the smallest number p satisfying this equation is called *period* of x. It is easy to see that for all $j \geqslant i$, we have $x^{j+p} = x^j$. In particular, we have $(x^{ip})^2 = x^{ip+ip} = x^{ip}$, which shows that in a finite semigroup, every element has an idempotent power. The idempotent power of an element $x \in S$ is unique. Taking the least common multiple of all such exponents, one obtains a natural number ω_S such that $x^{\omega_S} \in E(S)$ for all $x \in S$. When the reference to S is clear from the context, we skip the index and write ω instead of ω_S. The following elementary property shall be used later on.

Lemma 2.1. *Let S be a finite semigroup of cardinality n and let $s_1, \ldots, s_n \in S$. Then there exist an index $i \in \{1, \ldots, n\}$ and an idempotent element e such that $s_1 \cdots s_i e = s_1 \cdots s_i$. This idempotent element e can be written as a product over $\{s_1, \ldots, s_n\}$.*

Proof. For $1 \leqslant i \leqslant n$, let $p_i = s_1 \cdots s_i$. If the elements $p_1, \ldots, p_n$ are pairwise disjoint, one of them is idempotent and the statement holds. Otherwise, there exist integers $i, j \in \{1, \ldots, n\}$ with $i < j$ and with $p_i = p_j$. In this case, $p_i = p_i s_{i+1} \cdots s_j = p_i (s_{i+1} \cdots s_j)^\omega$ which yields the claim. □

Following classical terminology, a *band* is a semigroup where every element is idempotent. A *group* is a monoid whose only idempotent element is the neutral element 1. A *zero* element z of a finite semigroup S satisfies $zx = z = xz$ for all $x \in S$. If a zero element exists, it is often denoted by 0. Each semigroup contains at most one zero element and a semigroup is *nilpotent* if its only idempotent element is a zero element. An element $x \in S$ is *central* if it commutes with all other elements, i.e., $xy = yx$ for all $y \in S$. If all elements of a semigroup are central, the semigroup is *commutative*. A commutative band is also called *semilattice*.

A *subsemigroup* of a semigroup is a subset closed under multiplication. A *subgroup* of a semigroup is a subsemigroup which forms a group. For a semigroup S and a subset X of S, we denote by $\langle X \rangle$ the subsemigroup of S *generated by* X, i.e., the smallest subsemigroup of S containing X. The elements of the set X are called *generators* for this subsemigroup. A semigroup S is *cyclic* if it is generated by a singleton set. In this case, the generator x of S is unique, the *index of* S is the index of x and the *period of* S is the period of x. If S is a semigroup and p, q are elements of S, then the set

$pSq = \{psq \mid s \in S\}$ forms a subsemigroup of S. If $p = q = e$ for some idempotent element $e \in E(S)$, then eSe is a subsemigroup of S with neutral element e, called the *local monoid at e*.

Let S and T be semigroups and let M and N be monoids. The *direct product* of S and T is the Cartesian product $S \times T$ with componentwise multiplication. A *semigroup morphism* from S to T is a mapping $h\colon S \to T$ such that $h(s)h(t) = h(st)$ for all $s, t \in S$. A *monoid morphism* from M to N is a semigroup morphism $h\colon M \to N$ which additionally satisfies $h(1) = 1$. We often use the term *morphism* to refer to both semigroup and monoid morphisms if the reference is clear from the context. A semigroup T is a *divisor* of S if there exists a surjective semigroup morphism from a subsemigroup of S onto T. It is easy to verify that the division relation is transitive. We will also need the following result later in this section.

Lemma 2.2. *Let M be a monoid and let S be a semigroup which is not a monoid. Then S divides M if and only if S^1 divides M.*

Proof. Clearly, S is a subsemigroup of S^1, so it suffices to prove the direction from left to right. Suppose that S divides M, i.e., there exists a surjective morphism $h\colon T \to S$ where T is a subsemigroup of M. If $1 \in T$, then $h(1)$ is the identity element in S since $h(1)h(s) = h(s) = h(1)h(s)$ for all $s \in S$. This contradicts the assumption that S is not a monoid. Thus, we can extend h to a surjective monoid morphism from $T \cup \{1\}$ to S^1. □

A *left congruence* on a semigroup S is a relation $\sim$ on S such that $x \sim y$ implies $px \sim py$ for all $x, y, p \in S$. A *right congruence* is defined symmetrically. A *congruence* is a relation which is both a left and a right congruence. For a congruence $\sim$ on a semigroup S, one can define the *quotient* $S/{\sim}$ which is the set of equivalence classes of S modulo $\sim$ equipped with the canonical multiplication induced by the multiplication in S. The quotient again forms a semigroup.

2.2.2 Green's Relations and Local Theory

Green's relations are a useful tool to study structural properties of finite semigroups. For a finite semigroup S and elements $s, t \in S$ let

$$\begin{array}{ll}
s \leqslant_{\mathcal{R}} t \text{ if there exists } q \in S^1 \text{ such that } s = tq, & s \mathrel{\mathcal{R}} t \text{ if } s \leqslant_{\mathcal{R}} t \text{ and } t \leqslant_{\mathcal{R}} s, \\
s \leqslant_{\mathcal{L}} t \text{ if there exists } p \in S^1 \text{ such that } s = pt, & s \mathrel{\mathcal{L}} t \text{ if } s \leqslant_{\mathcal{L}} t \text{ and } t \leqslant_{\mathcal{L}} s, \\
s \leqslant_{\mathcal{J}} t \text{ if there exist } p, q \in S^1 \text{ such that } s = ptq, & s \mathrel{\mathcal{J}} t \text{ if } s \leqslant_{\mathcal{J}} t \text{ and } t \leqslant_{\mathcal{J}} s, \\
s \leqslant_{\mathcal{H}} t \text{ if } s \leqslant_{\mathcal{R}} t \text{ and } s \leqslant_{\mathcal{L}} t, & s \mathrel{\mathcal{H}} t \text{ if } s \leqslant_{\mathcal{H}} t \text{ and } t \leqslant_{\mathcal{H}} s.
\end{array}$$

The relation $\mathcal{R}$ (resp. $\mathcal{L}$, $\mathcal{J}$, $\mathcal{H}$) is an equivalence relation and its equivalence classes are called *$\mathcal{R}$-classes* (resp. *$\mathcal{L}$-classes*, *$\mathcal{J}$-classes*, *$\mathcal{H}$-classes*). It is straightforward to verify that $\mathcal{R}$ is a left congruence and $\mathcal{L}$ is a right congruence.

A semigroup is *$\mathcal{R}$-trivial* if all its $\mathcal{R}$-classes are singletons; $\mathcal{L}$-trivial and $\mathcal{J}$-trivial semigroups are defined analogously. The following theorem is a standard result in semigroup theory [Pin86, Alm94, RS09].

Theorem 2.3. *Let S be a finite semigroup and let $s, t \in S$. Then, the following properties are equivalent:*

1. *$s \mathrel{\mathcal{J}} t$.*
2. *There exists $u \in S$ such that $s \mathrel{\mathcal{R}} u \mathrel{\mathcal{L}} t$.*
3. *There exists $u \in S$ such that $s \mathrel{\mathcal{L}} u \mathrel{\mathcal{R}} t$.*

This theorem justifies the use of so called *egg-box diagrams* to depict the structure of finite semigroups. Such a diagram consists of boxes representing the $\mathcal{J}$-classes. Each box contains a grid in which each row corresponds to a $\mathcal{R}$-class and each column corresponds to an $\mathcal{L}$-class. A star is used to indicate that an $\mathcal{H}$-class contains an idempotent element. Each such $\mathcal{H}$-class forms a subgroup of the semigroup. Some examples are given in Figure 2.1.

If s is an element of a semigroup S and e is an idempotent element of S such that $s \leqslant_{\mathcal{R}} e$, then $s = eq$ for some $q \in S^1$ and thus, $es = eeq = eq = s$. Similarly, if e is idempotent and $s \leqslant_{\mathcal{L}} e$, then $se = s$. The following theorem of Clifford and Miller is another very useful link between the structure of semigroups and the existence of certain idempotent elements [Pin86, Alm94, RS09].

Theorem 2.4 (Location Theorem)**.** *Let S be a finite semigroup and let $s, t \in S$ with $s \mathrel{\mathcal{J}} t$. Then, the following properties are equivalent:*

1. *$st \mathrel{\mathcal{J}} s$.*
2. *$s \mathrel{\mathcal{R}} st \mathrel{\mathcal{L}} t$.*
3. *There exists an idempotent element $e \in E(S)$ such that $s \mathrel{\mathcal{L}} e \mathrel{\mathcal{R}} t$.*

A $\mathcal{J}$-class is *regular* if it contains an idempotent element. As an immediate consequence of the previous theorem, every $\mathcal{R}$-class and every $\mathcal{L}$-class of a regular $\mathcal{J}$-class contains at least one idempotent element, and every $\mathcal{H}$-class contains at most one idempotent element.

For a given semigroup, we will also consider the binary relations RM, LM, GGM and AGGM defined by

$$\begin{aligned}
s \mathrel{\mathsf{RM}} t \quad &\text{if} \quad \forall x\colon \rho(x) \to \big(\neg(xs \mathrel{\mathcal{J}} x \,\vee\, xt \mathrel{\mathcal{J}} x) \vee xs = xt\big),\\
s \mathrel{\mathsf{LM}} t \quad &\text{if} \quad \forall x\colon \rho(x) \to \big(\neg(sx \mathrel{\mathcal{J}} x \,\vee\, tx \mathrel{\mathcal{J}} x) \vee sx = tx\big),\\
s \mathrel{\mathsf{GGM}} t \quad &\text{if} \quad \forall x \forall y\colon (\rho(x) \,\wedge\, x \mathrel{\mathcal{J}} y) \to \big(\neg(xsy \mathrel{\mathcal{J}} x \,\vee\, xty \mathrel{\mathcal{J}} x) \vee xsy = xty\big),\\
s \mathrel{\mathsf{AGGM}} t \quad &\text{if} \quad \forall x \forall y\colon (\rho(x) \,\wedge\, x \mathrel{\mathcal{J}} y) \to \big(xsy \mathrel{\mathcal{J}} x \leftrightarrow xty \mathrel{\mathcal{J}} x\big),
\end{aligned}$$

where the abbreviation $\rho(x) = (\exists e\colon ee = e \wedge e \mathrel{\mathcal{J}} x)$ is used to express that the $\mathcal{J}$-class of x is regular. We have $s \mathrel{\mathsf{RM}} t$ (resp. $s \mathrel{\mathsf{LM}} t$) if and only if s and t define the same partial transformations on the right (resp. left) of each regular $\mathcal{J}$-class. It is not difficult to show that in any given finite semigroup, each of the relations RM, LM, GGM and AGGM is a congruence; see e.g. [RS09].

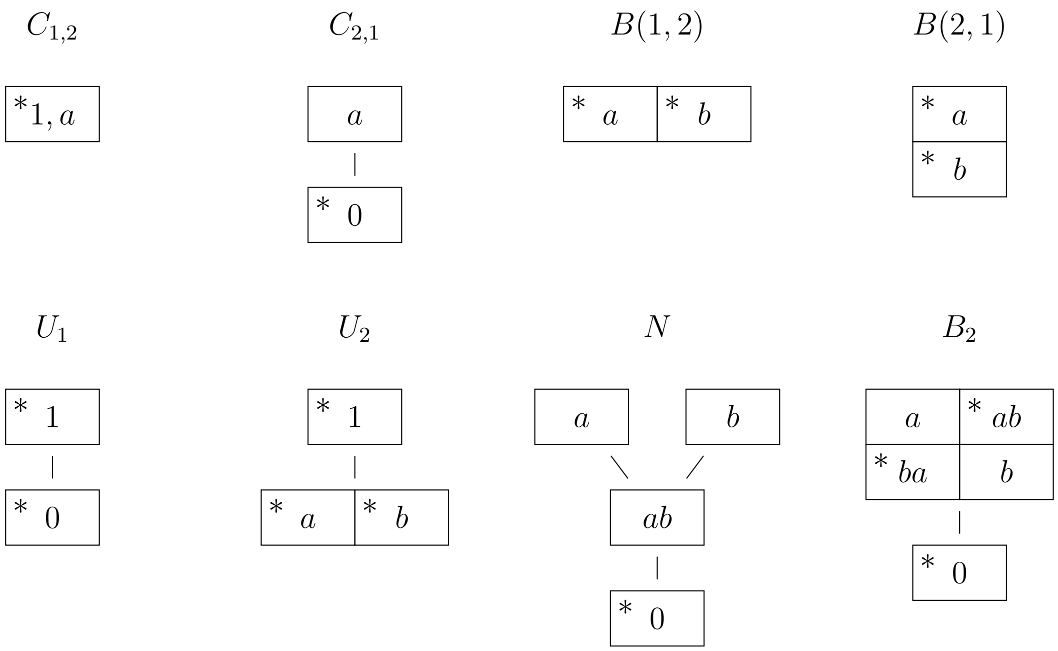


Figure 2.1: Egg-box diagrams of some of the semigroups defined in Section 2.2.3

2.2.3 Examples of Finite Semigroups

Below, we define some commonly occurring finite semigroups. These semigroups play an important role in the following chapters. We denote by $C_{i,p}$ the cyclic semigroup of index i and period p. This semigroup is a group if and only if $i = 1$. Its generator is usually denoted by the letter a.

Another simple family of semigroups are *Rees matrix semigroups*. These semigroups come in two flavors. Let G be a finite group (the so-called *structure group*) and let A, B be two disjoint finite sets (called *index sets*). For a mapping $C\colon B \times A \to G$ (called *sandwich matrix*), the Rees matrix semigroup $\mathcal{M}(G, A, B, C)$ is defined as the set $A \times G \times B$, equipped with the multiplication

$$(a_1, g_1, b_1)(a_2, g_2, b_2) = (a_1, g_1C(b_1, a_2)g_2, b_2)$$

Analogously, one can define Rees matrix semigroups for sandwich matrices with zero entries $C\colon B \times A \to G \cup \{0\}$. In this case, the Rees matrix semigroup $\mathcal{M}^0(G, A, B, C)$ is defined as the set $A \times G \times B \cup \{0\}$, equipped with the multiplication

$$(a_1, g_1, b_1)(a_2, g_2, b_2) = \begin{cases} (a_1, g_1C(b_1, a_2)g_2, b_2) & \text{if } C(b_1, a_2) \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

The element 0 is the zero element.

When G is the trivial group, the product depends only on the cardinalities of the index sets A and B (up to isomorphism). For $m, n \in \mathbb{N} \setminus \{0\}$, we denote by $B(m, n)$ the Rees matrix semigroup $\mathcal{M}(\{1\}, A, B, C)$ with a trivial structure group, with index sets $A = \{1, \dots, m\}$ and $B = \{1, \dots, n\}$ and with the constant sandwich matrix $C \colon B \times A \to \{1\}$. For example, the semigroup $B(1,1)$ is the trivial semigroup with a single zero element and $B(1,2)$ is isomorphic to $\{a, b\}$ with $a^2 = ba = a$ and $ab = b^2 = b$. The monoid U_1 is defined as $B(1,1)^1$; it consists of a neutral element 1 and a zero element 0. The monoid U_2 is defined as $B(1,2)^1$. For convenience, we will consistently use the letters a and b to refer to the non-neutral elements of $B(1,2)$ and U_2.

The semigroup $N = \{a, b, ab, 0\}$ with zero element 0 is defined by $a^2 = b^2 = ba = 0$. The semigroup $B_2 = \{a, b, ab, ba, 0\}$ is defined by $aba = a$, $bab = b$ and $a^2 = b^2 = 0$. The egg-box diagrams of some of the semigroups introduced above are depicted in Figure 2.1.

Another important family of semigroups are *transformation semigroups*. For a set Q, the set of all transformations $f \colon Q \to Q$ forms a monoid with function composition as binary operation. This monoid is called the *full transformation semigroup* or *full transformation monoid* on Q.

2.2.4 Varieties of Finite Semigroups

A *variety of finite semigroups* is a class of finite semigroups closed under finite direct products and under taking divisors. Note that in the literature, such classes of semigroups are often called *pseudovarieties*, as opposed to *Birkhoff varieties* which are also closed under infinite direct products. We will often use the term *varieties* to refer to varieties of finite semigroups.

Varieties are often defined using so-called ω-identities. For a set of variables X, the set of *ω-terms* over X is defined inductively as follows: every variable from X is an ω-term and if U and V are ω-terms, then so are UV and U^ω. An *ω-identity* is an expression of the form $U = V$ where U and V are ω-terms. If neither U nor V contain a subterm of the form W^ω, the identity is called an *equation*. Every mapping $h \colon X \to S$ to a finite semigroup S extends uniquely to ω-terms by $h(UV) = h(U)h(V)$ and $h(U^\omega) = (h(U))^{\omega_S}$. Such a mapping *satisfies* an ω-identity $U = V$ if $h(U) = h(V)$. A finite semigroup S satisfies an ω-identity if the identity is satisfied by every mapping $h \colon X \to S$. A set of ω-identities is satisfied if each of the identities in the set is satisfied. The class of finite semigroups *defined* by a set of ω-identities is the class of all finite semigroups satisfying the given set. It is well known that every class of finite semigroups defined by a (not necessarily finite) set of ω-identities is a variety [Rei82]. Table 2.1 gives an overview of some basic varieties occurring in this work, together with their defining ω-identities. Proofs for these ω-identities can be found in standard textbooks such as [Alm94, RS09].

Not all varieties of finite semigroups can be defined using ω-identities. For example, if $\mathbf{H}$ is a variety of finite groups which contains all nilpotent groups, then $\mathbf{H}$ cannot be defined by ω-identities unless $\mathbf{H} = \mathbf{G}$. This is an easy consequence of Baumslag's result that the free group is residually a finite p-group [Bau65]. It applies in particular to the variety $\mathbf{G}_{\text{sol}}$ of *solvable groups*, i.e., the variety of all finite groups having a normal series whose factor groups are all Abelian [Rot99]. We will briefly refer to solvable groups later.

Variety	ω-identities	Description
S	$x = x$	all finite semigroups
I	$x = y$	trivial semigroup(s)
Com	$xy = yx$	commutative semigroups
G	$x^\omega y = y = yx^\omega$	groups
CR	$x^{\omega+1} = x$	completely regular semigroups
A	$x^{\omega+1} = x^\omega$	aperiodic semigroups
Ab	$x^\omega y = y, xy = yx$	Abelian groups
J	$y(xy)^\omega = (xy)^\omega = (xy)^\omega x$	$\mathcal{J}$-trivial semigroups
R	$(xy)^\omega x = (xy)^\omega$	$\mathcal{R}$-trivial semigroups
L	$y(xy)^\omega = (xy)^\omega$	$\mathcal{L}$-trivial semigroups
B	$x^2 = x$	bands
$\mathbf{J_1}$	$x^2 = x, xy = yx$	semilattices
$\mathbf{R_1}$	$x^2 = x, xyx = xy$	left regular bands
$\mathbf{L_1}$	$x^2 = x, xyx = yx$	right regular bands
NB	$x^2 = x, xyzx = xzyx$	normal bands
RB	$x^2 = x, xyxzx = xyzx$	regular bands

Table 2.1: Varieties of finite semigroups and their defining ω-identities

The varieties in the lower half of Table 2.1 are varieties of bands. The lattice of band varieties was studied extensively by Birjukov, Fennemore and Gerhard [Bir70, Fen71, Ger70]. The classes in Table 2.1 are only a small excerpt of this lattice.

The relation between these varieties is depicted in Figure 2.2. There are many alternative characterizations of these varieties. For example, $\mathbf{R_1}$ is the variety of all finite $\mathcal{R}$-trivial bands. An easy way to see this is combining the ω-identity $(xy)^\omega x = (xy)^\omega$ for **R** with the fact that in bands, the ω-operator is the identity. Similarly, $\mathbf{J_1}$ is the variety of all finite $\mathcal{J}$-trivial bands and $\mathbf{L_1}$ is the variety of all finite $\mathcal{L}$-trivial bands. We will see later that the variety $\mathbf{J_1}$ (resp. $\mathbf{R_1}$, $\mathbf{L_1}$) is the smallest variety of finite semigroups containing U_1 (resp. U_2, U_2^{op}); see Proposition 2.7 for details.

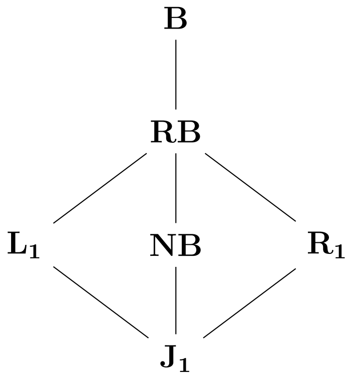


Figure 2.2: Varieties of finite bands

One can show that **RB** is the smallest variety of finite semigroups containing both $\mathbf{R_1}$ and $\mathbf{L_1}$. It is also the largest variety of finite bands where Green's relations $\mathcal{R}$ and $\mathcal{L}$ are congruences. For proofs and further results on **RB**, we refer to [Pet77]. Note that, as opposed to our definition, in the literature, the identifier **RB** is sometimes also used to refer to the class of *rectangular bands* which is a strict subclass of **NB**.

The class of normal bands is the largest variety of finite semigroups whose monoids are semilattices. This follows from the following alternative characterizations of **NB** which were originally established by Yamada [Yam67].

Proposition 2.5. *Let S be a finite band. Then, the following properties are equivalent:*

(1) S is a normal band.

(2) For all $p, q \in S$, the set pSq is a commutative subsemigroup of S.

(3) All local monoids of S are semilattices.

Proof. (1) implies (2). Suppose that S is a normal band and let $p, q \in S$. It is clear that the set pSq forms an idempotent subsemigroup of S. Moreover, we have $psq\ ptq = psq\ ptq\ ptq = p(psq)(ptq)ptq = p(ptq)(psq)ptq = ptq(psq)(ptq)q = ptq(ptq)(psq)q = ptq\ psq$ which shows that pSq is commutative.

(2) implies (3) is trivial.

(3) implies (1). Suppose that S is a band, all of whose local monoids are semilattices. We want to verify that, for all $x, y, z \in S$, we have $xyzx = xzyx$. First, note that

$$\begin{aligned} xyzx &= xyxyxy\ zxzxzx = (xyx)(xyx)(xyzx)(xzx)(xzx) \\ &= (xyx)(xzx)(xyzx)(xyx)(xzx) = xyx\ zxy\ zxy\ xzx \\ &= xyx\ zxy\ xzx = xyxzx\ xyxzx = xyxzx \end{aligned}$$

where the third equality uses commutativity of xSx and the remaining equalities use that S is a band. Similarly, we obtain $xzyx = xzxyx$ and thus, $xyzx = xyxzx = (xyx)(xzx) = (xzx)(xyx) = xzxyx = xzyx$, as desired. □

2.2.5 Varieties Defined by Operations

Varieties can also be defined as the result of certain operations. For example, if **V** and **W** are varieties, it is easy to see that their intersection $\mathbf{V} \cap \mathbf{W}$ is a variety as well. On the other hand, the union of two varieties is not necessarily a variety.

The variety *generated* by a class of finite semigroups **C** is the smallest variety containing **C**. It is the closure of **C** under direct products and divisors. We will consider some varieties generated by a single semigroup, based on the following lemma.

Lemma 2.6. *Let $u, v \in A^+$ with $u \neq v$.*

1. *If $\mathrm{alph}(u) \neq \mathrm{alph}(v)$, there exists a morphism $h\colon A^+ \to U_1$ such that $h(u) \neq h(v)$.*

2. *If $|u|_a \leqslant 1$ and $|v|_a \leqslant 1$ for all $a \in A$, there exists a morphism $h\colon A^+ \to U_2$ such that $h(u) \neq h(v)$.*

3. *If u and v are of the form $a_1^2 \cdots a_k^2 b_1 \cdots b_\ell$ with $a_1 < \cdots < a_k$ for some (common) strict linear order $<$ on A such that the letters $a_1, \ldots, a_k, b_1, \ldots, b_\ell \in A$ are pairwise disjoint, there exists a morphism $h\colon A^+ \to N^1$ such that $h(u) \neq h(v)$.*

Proof. (1) By symmetry, we may assume that there exists a letter $a \in \mathrm{alph}(u) \setminus \mathrm{alph}(v)$. Let $h\colon A^+ \to U_1$ be the morphism defined by $h(a) = 0$ and $h(c) = 1$ for all $c \in A \setminus \{a\}$. Clearly, $h(u) = 0 \neq 1 = h(v)$.

(2) If $\mathrm{alph}(u) \neq \mathrm{alph}(v)$, we can use the fact that the subsemigroup $\{1, a\}$ of U_2 is isomorphic to U_1 and then use (1) to obtain a morphism which distinguishes u from v. Otherwise, we may factorize $u = u_1aw$ and $v = v_1bw$ with $u_1, v_1, w \in A^*$ and $a, b \in A$ such that $a \neq b$. Note that since $|u|_a = |v|_b = 1$, neither a nor b appears in the word w. Thus, the morphism $h\colon A^+ \to U_2$ defined by $h(a) = a$, $h(b) = b$ and $h(c) = 1$ for $c \in A \setminus \{a, b\}$ satisfies $h(u) = h(u_1a)h(w) = a \neq b = h(v_1b)h(w) = h(v)$.

(3) If there exists some letter $a \in A$ such that $|u|_a \neq |v|_a$, then setting $h(a) = a$ and $h(c) = 1$ for $c \in A \setminus \{a\}$, we obtain $h(u) \neq h(v)$. Thus, we may assume that $|u|_a = |v|_a$ for all $a \in A$. In particular, the prefix $a_1^2 \cdots a_k^2$ of letters occurring twice in u or in v is the same. Since $u \neq v$, we can factorize $u = u_1aw$ and $v = v_1bw$ with $u_1, v_1, w \in A^*$ and $a \neq b \in A$ such that the word u_1w does not contain the letter a while v_1w does not contain the letter b. In particular, the word w contains neither a nor b. By the assumption that $|u|_a = |v|_a$ and $|u|_b = |v|_b$, we obtain that u_1 contains exactly one b and v_1 contains exactly one a. Thus, setting $h(a) = a$, $h(b) = b$ and $h(c) = 1$ for $c \in A \setminus \{a, b\}$, we obtain $h(u) = h(u_1a)h(w) = ba = 0 \neq ab = h(v_1b)h(w) = h(v)$. □

This lemma allows us to derive equations for the varieties generated by the monoids U_1, U_2, U_2^{op} and N^1.

Proposition 2.7. *The variety* $\mathbf{J_1}$ *is generated by the monoid* U_1*, the variety* $\mathbf{L_1}$ *is generated by the monoid* U_2 *and* $\mathbf{R_1}$ *is generated by the monoid* U_2^{op}*. The variety generated by* N^1 *is defined by the equations* $x^3 = x^2$ *and* $x^2y = xyx = yx^2$*.*

Proof. We will only prove the statements for U_1, U_2 and for N^1. The statement for U_2^{op} follows by left-right symmetry. A routine calculation shows that U_1 is a semilattice, U_2 is idempotent and satisfies the equation $xyx = yx$ and that N^1 satisfies both $x^3 = x^2$ and $x^2 = xyx = yx^2$. It remains to prove that every semigroup satisfying both $x^2 = x$ and $xy = yx$ (resp. both $x^2 = x$ and $xyx = yx$, both $x^3 = x^2$ and $x^2y = xyx = yx^2$) belongs to the variety generated by U_1 (resp. U_2, N^1).

Let S be a finite semigroup and let A be a set of generators. Let $<$ be some fixed strict linear order on A. Two words $u, v \in A^+$ are said to be *equivalent* if they evaluate to the same element in S.

If S is a semilattice, every word over A can be transformed into an equivalent word of the form $a_1 \cdots a_k$ with $a_1 < \cdots < a_k$ by repeatedly applying the equations $xy = yx$ and $x^2 = x$. Thus, for each pair of non-equivalent words $u, v \in A^+$, there exists a morphism $h\colon A^+ \to U_1$ such that $h(u) \neq h(v)$ by Lemma 2.6. This implies that S divides a finite direct product of copies of U_1.

If S satisfies $x^2 = x$ and $xyx = yx$, every word over A can be transformed to an equivalent word in which no letter appears more than once. Therefore, for each pair of non-equivalent words $u, v \in A^+$, there exists a morphism $h\colon A^+ \to U_2$ such that $h(u) \neq h(v)$ by Lemma 2.6. Thus, S divides a finite direct product of copies of U_2.

If S satisfies both $x^3 = x^2$ and $x^2y = xyx = yx^2$, every word over A can be transformed to an equivalent word which has the properties described in item (3) of Lemma 2.6. Therefore, for each pair of non-equivalent words $u, v \in A^+$, there exists a morphism $h\colon A^+ \to N^1$ such that $h(u) \neq h(v)$. In particular, S divides a finite direct product of copies of N^1. $\square$

The first part of the previous proposition has another immediate consequence on varieties of finite monoids.

Corollary 2.8. *Let $\mathbf{V}$ be a variety of finite monoids. Then, either $\mathbf{V} \subseteq \mathbf{G}$ or $\mathbf{J_1} \subseteq \mathbf{V}$.*

Proof. Clearly, the only group which is also a semilattice is the trivial group. Suppose now that $\mathbf{V}$ is a variety of finite monoids such that $\mathbf{V} \not\subseteq \mathbf{G}$. Let $M \in \mathbf{V} \setminus \mathbf{G}$. Then, M contains an idempotent element e which is not the neutral element. The subsemigroup $\{1, e\}$ of M is isomorphic to U_1. Thus, $U_1 \in \mathbf{V}$ which yields $\mathbf{J_1} \subseteq \mathbf{V}$ by the previous proposition. $\square$

Using techniques similar to those used in Lemma 2.6 and Proposition 2.7, one can characterize the variety generated by B_2 in terms of equations. Since the proof is more technical, we only state the result here and refer to [Tis80, Tra81] for a full proof.

Theorem 2.9 (Tishchenko, Trahtman)**.** *The variety generated by B_2 is defined by the equations $x^3 = x^2$, $x^2y^2 = y^2x^2$ and $xyxyx = xyx$.*

Another very common way of defining varieties are unary operators. For a variety of finite semigroups $\mathbf{V}$, we will denote by

- $\mathbb{D}\mathbf{V}$ the class of all finite semigroups whose regular $\mathcal{J}$-classes are subsemigroups which belong to $\mathbf{V}$,
- $\mathbb{E}\mathbf{V}$ the class of all finite semigroups whose maximal idempotent-generated subsemigroups $\langle E(S) \rangle$ belong to $\mathbf{V}$,
- $\mathbb{L}\mathbf{V}$ the class of all finite semigroups whose local monoids belong to $\mathbf{V}$,
- $\mathbf{V}^{\mathbf{N}}$ the class of all finite semigroups S such that the subsemigroup $SE(S)S$ belongs to $\mathbf{V}$,
- $\mathbf{K} \mathbin{\textcircled{m}} \mathbf{V}$ the class of all finite semigroups S such that S/RM belongs to $\mathbf{V}$,
- $\mathbf{D} \mathbin{\textcircled{m}} \mathbf{V}$ the class of all finite semigroups S such that S/LM belongs to $\mathbf{V}$,
- $\mathbf{N} \mathbin{\textcircled{m}} \mathbf{V}$ the class of all finite semigroups S such that $S/(\mathsf{RM} \cap \mathsf{LM})$ belongs to $\mathbf{V}$,
- $\mathbb{L}\mathbf{I} \mathbin{\textcircled{m}} \mathbf{V}$ the class of all finite semigroups S such that S/GGM belongs to $\mathbf{V}$, and
- $\mathbb{L}\mathbf{G} \mathbin{\textcircled{m}} \mathbf{V}$ the class of all finite semigroups S such that S/AGGM belongs to $\mathbf{V}$.

Variety	ω-identities	Description
$\mathbb{D}\mathbf{G}$	$(xy)^\omega = (yx)^\omega$	regular $\mathcal{J}$-classes are groups
$\mathbb{D}\mathbf{S}$	$((xy)^\omega(yx)^\omega(xy)^\omega)^\omega = (xy)^\omega$	regular $\mathcal{J}$-classes are semigroups
$\mathbb{L}\mathbf{I}$	$x^\omega y x^\omega = x^\omega$	locally trivial semigroups
$\mathbb{L}\mathbf{G}$	$(x^\omega y x^\omega)^\omega = x^\omega$	locally groups
$\mathbf{G}^{\mathbf{N}}$	$x^\omega = y^\omega$	exactly one idempotent element

Table 2.2: Varieties defined by the operations $\mathbb{D}\mathbf{V}$, $\mathbb{L}\mathbf{V}$ and $\mathbf{V}^{\mathbf{N}}$

The classes $\mathbf{K} \mathbin{\textcircled{m}} \mathbf{V}$, $\mathbf{D} \mathbin{\textcircled{m}} \mathbf{V}$, $\mathbf{N} \mathbin{\textcircled{m}} \mathbf{V}$, $\mathbb{L}\mathbf{I} \mathbin{\textcircled{m}} \mathbf{V}$ and $\mathbb{L}\mathbf{G} \mathbin{\textcircled{m}} \mathbf{V}$ are *Mal'cev products* and are usually defined using *relational morphisms.* For our purposes, the definitions above will suffice. For a variety of finite groups $\mathbf{H}$, we denote by $\overline{\mathbf{H}}$ the class of all finite semigroups whose subgroups belong to $\mathbf{H}$. One can show that whenever starting with a variety of finite semigroups $\mathbf{V}$ (resp. variety of finite groups $\mathbf{H}$), each of the classes $\mathbb{D}\mathbf{V}$, $\mathbb{E}\mathbf{V}$, $\mathbb{L}\mathbf{V}$, $\mathbf{V}^{\mathbf{N}}$, $\mathbf{K} \mathbin{\textcircled{m}} \mathbf{V}$, $\mathbf{D} \mathbin{\textcircled{m}} \mathbf{V}$, $\mathbf{N} \mathbin{\textcircled{m}} \mathbf{V}$, $\mathbb{L}\mathbf{I} \mathbin{\textcircled{m}} \mathbf{V}$ and $\mathbb{L}\mathbf{G} \mathbin{\textcircled{m}} \mathbf{V}$ (resp. $\overline{\mathbf{H}}$) form varieties of finite semigroups [Alm94, RS09]. Table 2.2 gives an overview of some varieties obtained by applying some of these operators to varieties from Table 2.1 and also provides ω-identities. For proofs of these identities, we refer to [Alm94, RS09]. The variety $\mathbb{L}\mathbf{I}$ contains an infinite strict hierarchy $(\mathbb{L}\mathbf{I}_k)_{k \geqslant 1}$ of subvarieties where $\mathbb{L}\mathbf{I}_k$ is defined by the equation $x_1 \cdots x_k z y_k \cdots y_1 = x_1 \cdots x_k y_k \cdots y_1$; see [Alm94] for details.

An example for an application of the $\mathbb{E}$-operator is the variety $\mathbb{E}\mathbf{A}$. This variety cannot be defined using finitely many ω-identities, a fact that has originally been proved by Volkov [Vol95] and also follows from results proved in this thesis by an entirely different technique; see Corollary 5.47.

Lastly, one can define varieties of finite semigroups by *exclusion.* The next lemma shows how the location of idempotent elements within a $\mathcal{J}$-class yields non-trivial divisors.

Lemma 2.10. *Let S be a finite semigroup and let $e, f \in E(S)$ with $e \neq f$.*

1. *If $e \mathrel{\mathcal{R}} f$, then $B(1,2)$ divides S.*

2. *If $e \mathrel{\mathcal{L}} f$, then $B(2,1)$ divides S.*

3. *If $ef \mathrel{\not\mathcal{J}} e \mathrel{\mathcal{J}} f \mathrel{\not\mathcal{J}} fe$, then B_2 divides S.*

4. *If $ef \mathrel{\not\mathcal{J}} e \mathrel{\mathcal{J}} f$, then B_2 divides $S \times S$.*

Proof. (1) Let T be the subsemigroup of S generated by $\{e, f\}$. Since $e \mathrel{\mathcal{R}} f$, we have $ef = f$ and $fe = e$, which shows that T is isomorphic to $B(1,2)$.

(2) is dual to (1).

(3) By Theorem 2.3, there exist elements $s, t \in S$ with $e \mathrel{\mathcal{R}} s \mathrel{\mathcal{L}} f$ and with $e \mathrel{\mathcal{L}} t \mathrel{\mathcal{R}} f$. Let $\overline{s} = (ts)^{\omega-1}t$. By Theorem 2.4, we obtain $s\overline{s} = (st)^\omega \mathrel{\mathcal{H}} e$ and $\overline{s}s = (ts)^\omega \mathrel{\mathcal{H}} f$. Since every $\mathcal{H}$-class contains at most one idempotent element, this implies $s\overline{s} = e$ and

$\overline{s}s = f$. Using $ef \mathrel{\not\mathcal{J}} e$ and $fe \mathrel{\not\mathcal{J}} e$, Theorem 2.4 yields that there is no idempotent element in the $\mathcal{H}$-class of s nor in the $\mathcal{H}$-class of $\overline{s}$, and we can apply the theorem again to obtain that neither s^2 nor $\overline{s}^2$ belong to the $\mathcal{J}$-class of e.

Let T be the subsemigroup of S generated by $\{s, \overline{s}\}$ and let $h\colon T \to B_2$ be the morphism defined by $h(s) = a$ and $h(\overline{s}) = b$. Any element of T which can be written as a product containing the factor s^2 or $\overline{s}^2$ is mapped to the zero element of B_2. We have $h(e) = h(s\overline{s}) = ab$ and $h(f) = h(\overline{s}s) = ba$. Moreover, $h(s\overline{s}s) = aba = a = h(es) = h(s)$ and $h(\overline{s}s\overline{s}) = bab = b = h(\overline{s}e) = \overline{s}$. Therefore, h is well-defined and surjective.

(4) Suppose that $ef \mathrel{\not\mathcal{J}} e \mathrel{\mathcal{J}} f$. Then, in the direct product $S \times S$, we have $(e, f) \mathrel{\mathcal{J}} (f, e)$ but $(ef, fe) \mathrel{\not\mathcal{J}} (fe, ef)$ and $(fe, ef) \mathrel{\not\mathcal{J}} (ef, fe)$. Thus, the semigroup B_2 divides $S \times S$ by (3). □

Several examples of varieties defined by exclusion are given in the following theorem. The proofs are mostly based on the previous lemma.

Theorem 2.11. *Let S be a finite semigroup. Then,*

1. *S belongs to $\mathbf{CR}$ if and only if $C_{2,1}$ does not divide S.*
2. *S belongs to $\mathbb{D}\mathbf{S}$ if and only if B_2 does not divide $S \times S$.*
3. *S belongs to $\mathbb{D}\mathbf{G}$ if and only if neither B_2, nor $B(1, 2)$ nor $B(2, 1)$ divide S.*
4. *S belongs to $\mathbb{L}\mathbf{I}$ if and only if no non-trivial monoid divides S.*
5. *S belongs to $\mathbb{L}\mathbb{D}\mathbf{S}$ if and only if B_2^1 does not divide $S \times S$.*

Proof. It is easy to verify that $C_{2,1}$ does not belong to $\mathbf{CR}$, that B_2 does not belong to $\mathbb{D}\mathbf{S}$, that $B(1, 2)$ and $B(2, 1)$ do not belong to $\mathbb{D}\mathbf{G}$ and that B_2^1 does not belong to $\mathbb{L}\mathbb{D}\mathbf{S}$. Moreover, the variety $\mathbb{L}\mathbf{I}$ does not contain any non-trivial monoid, since setting $x = 1$ in the ω-identity $x^\omega y x^\omega = x^\omega$ yields the equation $y = 1$. Therefore, it suffices to prove the implications from right to left.

(1) Suppose that $S \notin \mathbf{CR}$ and let $s \in S$ such that $s^{\omega+1} \neq s$. Note that if the index i of s were 1 and the period of s is p, then $s^{\omega+1} = s^{p+1} = s$. Thus, the index of s is at least 2 and the morphism $h\colon \langle s \rangle \to C_{2,1}$ defined by $h(s) = a$ is well-defined.

(2) If $S \notin \mathbb{D}\mathbf{S}$, there exists a regular $\mathcal{J}$-class J of S and there exist elements $s, t \in J$ such that $st \notin J$. By Theorem 2.4 and by the fact that every $\mathcal{R}$- and every $\mathcal{L}$-class of a regular $\mathcal{J}$-class contains at least one idempotent element, we know that there also exist $e, f \in E(S)$ with $e \mathrel{\mathcal{J}} f$ but $ef \mathrel{\not\mathcal{J}} e$. Item (4) of Lemma 2.10 yields the desired statement.

(3) If $S \notin \mathbb{D}\mathbf{G}$, there exists a $\mathcal{J}$-class of S which contains at least two different idempotent elements e and f. If $ef \mathrel{\mathcal{J}} e$, then there exists an idempotent element $g \in E(S)$ with $e \mathrel{\mathcal{L}} g \mathrel{\mathcal{R}} f$ by Theorem 2.4. Since $e \neq f$, either $f \neq g$ or $e \neq g$. In the first case, $B(1, 2)$ divides S and in the second case, $B(2, 1)$ divides S by Lemma 2.10. By symmetry, either $B(1, 2)$ or $B(2, 1)$ divides S in case $fe \mathrel{\mathcal{J}} f$. The remaining case is $ef \mathrel{\not\mathcal{J}} e \mathrel{\mathcal{J}} f \mathrel{\not\mathcal{J}} fe$, whence B_2 divides S by Lemma 2.10.

(4) If $S \notin \mathbb{L}\mathbf{I}$, there exists some idempotent element $e \in E(S)$ such that the local monoid eSe is non-trivial. This monoid is a subsemigroup of S and thus yields a non-trivial divisor.

(5) If $S \notin \mathbb{L}\mathbb{D}\mathbf{S}$, there exists some $e \in E(S)$ such that the local monoid eSe does not belong to $\mathbb{D}\mathbf{S}$. By (2), the semigroup B_2 divides the monoid $eSe \times eSe$. Since B_2 is not a monoid, we can apply Lemma 2.2 to obtain that B_2^1 divides $eSe \times eSe$. □

For two varieties $\mathbf{V}$ and $\mathbf{W}$, the variety generated by $\mathbf{V} \cup \mathbf{W}$ is called the *join* of $\mathbf{V}$ and $\mathbf{W}$, denoted by $\mathbf{V} \vee \mathbf{W}$. It contains all divisors of direct products $S \times T$ where $S \in \mathbf{V}$ and $T \in \mathbf{W}$. The join of finite groups and finite semilattices is the variety of finite *Clifford semigroups*. Alternatively, this variety can be characterized as the intersection of $\mathbf{CR}$ and $\mathbb{D}\mathbf{G}$, using ω-identities, and using exclusions. We provide a minimal set of ω-identities as well as an additional identity that will be useful later.

Theorem 2.12. *Let S be a finite semigroup. The following properties are equivalent:*

1. *S belongs to $\mathbf{G} \vee \mathbf{J_1}$,*
2. *S satisfies the identities $(xy)^\omega = x^\omega y^\omega$, $x^\omega y = yx^\omega$ and $x^{\omega+1} = x$,*
3. *S satisfies the identities $x^\omega y = yx^\omega$, $x^{\omega+1} = x$,*
4. *S belongs to $\mathbf{CR} \cap \mathbb{D}\mathbf{G}$,*
5. *Neither $B(1,2)$ nor $B(2,1)$ nor $C_{2,1}$ divide S.*

Proof. (1) implies (2). Clearly, every group and every semilattice satisfies the ω-identities $x^\omega y = yx^\omega$ and $x^{\omega+1} = x$. Since these ω-identities form a variety, this also holds for every divisor of a direct product of groups and semilattices.

(2) implies (3) is trivial.

(3) implies (4). Suppose that s, t are elements of a regular $\mathcal{J}$-class such that $s \mathrel{\mathcal{R}} t$. Then, there exists some idempotent element $e \mathrel{\mathcal{R}} s$, and we have $es = s$. The ω-identity $x^\omega y = yx^\omega$ yields $se = s$, thus $e \mathrel{\mathcal{L}} s$ by Theorem 2.4. By the same argument, $e \mathrel{\mathcal{L}} t$, and therefore $s \mathrel{\mathcal{H}} e \mathrel{\mathcal{H}} t$. This means that every regular $\mathcal{J}$-class is a subgroup of S.

(4) implies (5) follows immediately from Theorem 2.11 and (5) implies (4) follows from the same theorem with the additional observation that $C_{2,1}$ divides B_2.

(4) implies (3). Let $e \in E(S)$ and let $s \in S$. Then $es = (es)^{\omega+1} = (es)^\omega es = (se)^\omega es = (se)^\omega s = s(es)^\omega = se(es)^\omega = se(se)^\omega = (se)^{\omega+1} = se$.

(3) implies (2). Let $s, t \in S$. Note that we have $s^\omega t^\omega \mathrel{\mathcal{R}} (st)^\omega$ since $s^\omega t^\omega (st)^\omega = s^\omega (st)^\omega t^\omega = (st)^\omega$ and $(st)^\omega (t^{\omega-1} s^{\omega-1})^\omega = s^\omega t^\omega$. For the last equality, note that we can repeatedly apply $(st)(t^{\omega-1} s^{\omega-1}) = st^\omega s^{\omega-1} = ss^{\omega-1} t^\omega = s^\omega t^\omega$ and move the factor $s^\omega t^\omega$ to the right after each step. Equivalently, $s^\omega t^\omega \mathrel{\mathcal{L}} (st)^\omega$. Also note that $s^\omega t^\omega$ is idempotent. Since each $\mathcal{H}$-class contains at most one idempotent element, this means that $s^\omega t^\omega = (st)^\omega$.

(2) implies (1). It suffices to show that for every surjective morphism $h\colon A^+ \to S$ and every pair of words $u, v \in A^+$ with $h(u) \neq h(v)$, there exists a morphism $g\colon A^+ \to T$ such that $g(u) \neq g(v)$ and $T \in \mathbf{J_1} \cup \mathbf{G}$ and $|T| \leqslant |S|$.

If $h(u) \not\mathrel{\mathcal{H}} h(v)$, we let $T = E(S)$ and we let $g\colon A^+ \to T$ be the morphism defined by $g(a) = (h(a))^\omega$ for all $a \in A$. The ω-identity $(xy)^\omega = x^\omega y^\omega$ implies that T is a subsemigroup of S and that g is well-defined. Moreover, the assumption $h(u) \not\mathrel{\mathcal{H}} h(v)$ implies $(h(u))^\omega \neq (h(v))^\omega$ and therefore, $g(u) = (h(u))^\omega \neq (h(v))^\omega = g(v)$. Since $x^\omega y = yx^\omega$, we obtain that T is a semilattice.

If $h(u) \mathrel{\mathcal{H}} h(v)$, let $e = (h(u))^\omega$ and let T be the $\mathcal{H}$-class of e. Note that T is a subgroup of S with neutral element e. We define a morphism $g\colon A^+ \to T$ by setting

$$g(a) = \begin{cases} h(a)e & \text{if } h(a)e \mathrel{\mathcal{H}} e, \\ e & \text{otherwise} \end{cases}$$

for all $a \in A$. For any word $w = a_1 \cdots a_\ell$ with $a_1, \ldots, a_\ell \in A$, note that we have $g(w) = h(a_1)e \cdots h(a_\ell)e = h(a_1 \cdots a_\ell)e$ which equals $h(w)$ if and only if $h(w) \mathrel{\mathcal{H}} e$. In particular, $g(u) \neq g(v)$. □

2.3 Recognizable Languages and Descriptional Complexity

In classical terminology, the class of languages definable by rational expressions is often referred to as the class of *regular languages*. Some text books use the term *rational languages* instead. The class of languages recognized by deterministic finite automata is then referred to as *recognizable languages*. While the classes of rational languages and recognizable languages coincide over finite words, i.e., the free semigroup over a finite alphabet, this does not necessarily hold for other structures. We therefore adopt the convention of using the term *recognizable languages* for languages recognized by deterministic finite automata or by finite semigroups.

A *deterministic finite automaton* (*DFA* for short) is a 5-tuple $(Q, A, \cdot, q_0, F)$ where Q is a finite set of *states*, A is a finite alphabet, $\cdot : Q \times A \to Q$ is the *transition function*, $q_0 \in Q$ is the *initial state* and $F \subseteq Q$ is the set of *accepting states*. We usually use the notation $q \cdot a$ instead of $\cdot\,(q, a)$. The transition function can be extended to words by setting $q \cdot a_1 \cdots a_\ell = (q \cdot a_1) \cdot a_2 \cdots a_\ell$ for $a_1, \ldots, a_\ell \in A$. The language *recognized* (or *accepted*) by the automaton is the set of all words $w \in A^+$ such that $q_0 \cdot w \in F$.

A morphism $h\colon A^+ \to S$ to a finite semigroup S *recognizes* a language $L \subseteq A^+$ if $h^{-1}(P) = L$ for some set $P \subseteq S$. The set P is often called the *accepting set* for L. By extension, we say that a semigroup S recognizes a language $L \subseteq A^+$ if there exists a morphism $h\colon A^+ \to S$ recognizing L. For a variety of finite semigroups $\mathbf{V}$ and an alphabet A, the class of all subsets of A^+ recognized by semigroups from $\mathbf{V}$ is denoted by $\mathbf{V}(A^+)$.

It is well-known that a language is recognized by a DFA if and only if it is recognized by a finite semigroup. For the direction from right to left, note that for a given morphism $h\colon A^+ \to S$ to a finite semigroup S and a set $P \subseteq S$, the automaton $(S^1, A, \cdot, 1, P)$ with $s \cdot a = sh(a)$ is easily seen to recognize $h^{-1}(a)$.

Conversely, note that for a given DFA $(Q, A, \cdot, q_0, F)$, every word $u \in A^+$ can be viewed as a transformation $u\colon Q \to Q$ by means of its right action on each state. These transformations all lie within the subsemigroup of the full transformation semigroup on Q generated by the letters in A. It is easy to see that the language recognized by the automaton is also recognized by this semigroup.

The conversion of semigroups to automata does not involve any significant blow-up: a semigroup with n elements is converted to an automaton with $n + 1$ states; see Section 2.4.5 for details on the encoding of these objects. However, the construction of a transformation semigroup from an automaton with n states yields a semigroup of size up to n^n. One can show that this upper bound is tight and is reached by a family of DFAs with only three letters; see e.g. [HK04]. The conversion also yields semigroups with inherently large numbers of $\mathcal{J}$-, $\mathcal{R}$-, $\mathcal{L}$- and $\mathcal{H}$-classes and exponentially long descending $<_{\mathcal{R}}$- and $<_{\mathcal{L}}$-chains [FK18b].

Algebraic properties are preserved in the sense that after converting a semigroup S to an automaton, going back to the transformation semigroup yields a semigroup isomorphic to the original semigroup S. Therefore, complexity upper bounds for automata usually imply the same bounds for semigroups and lower bounds for semigroups imply the same bounds for automata, even when algebraic restrictions are imposed on the input. However, the converse directions do not hold. Indeed, many decision problems turn out to be much easier when inputs are given as semigroups instead of DFAs. On the other hand, we will see that in some settings switching between automata and semigroups does not impact the complexity. This often indicates that hardness of a problem is caused by local structural properties.

2.4 Complexity Theory

We assume familiarity with standard definitions from complexity theory.

2.4.1 Random-Access Turing Machines

Classically, complexity classes are often defined using *sequential Turing machines* with sequential input tapes and sequential work tapes. Such a machine requires linear time to access the whole input. Since we are interested in algorithms running in sublinear time, we follow the convention of using *random-access Turing machines*. Such a Turing machine has a read-only input tape, a fixed number of sequential read-write work tapes, and a special sequential read-write *address tape* of length $\lceil \log n \rceil$. For simplicity, we only use binary tapes. In each time step, the machine has access to the input bit at the position currently encoded (in binary) on the address tape (or to the fact that the encoded position exceeds the input length). Configurations of a machine consist of the current state and the contents and positions of the read-write heads on each of the work tapes, including the address tape. The input is usually not denoted as part of the configuration but needs to be considered to decide whether a transition on a configuration reading an input bit is valid. For a function $f\colon \mathbb{N} \to \mathbb{N}$, we write

- $\mathsf{DTIME}(f(n))$ to denote the class of all languages decidable by a deterministic random-access Turing machine in time $\mathcal{O}(f(n))$,
- $\mathsf{NTIME}(f(n))$ to denote the class of all languages decidable by a non-deterministic random-access Turing machine in time $\mathcal{O}(f(n))$,
- $\mathsf{DSPACE}(f(n))$ to denote the class of all languages decidable by a deterministic random-access Turing machine with work tapes of size $\mathcal{O}(f(n))$, and
- $\mathsf{NSPACE}(f(n))$ to denote the class of all languages decidable by a non-deterministic random-access Turing machine with work tapes of size $\mathcal{O}(f(n))$.

We will also use the following abbreviations:

$$\begin{aligned}
\mathsf{DLOGTIME} &= \mathsf{DTIME}(\log n), & \mathsf{NLOGTIME} &= \mathsf{NTIME}(\log n),\\
\mathsf{DPOLYLOGTIME} &= \bigcup_{c\in\mathbb{N}} \mathsf{DTIME}(\log^c n), & \mathsf{NPOLYLOGTIME} &= \bigcup_{c\in\mathbb{N}} \mathsf{NTIME}(\log^c n),\\
\mathsf{L} &= \mathsf{DSPACE}(\log n), & \mathsf{NL} &= \mathsf{NSPACE}(\log n),\\
\mathsf{P} &= \bigcup_{c\in\mathbb{N}} \mathsf{DTIME}(n^c), & \mathsf{NP} &= \bigcup_{c\in\mathbb{N}} \mathsf{NTIME}(n^c),\\
\mathsf{PSPACE} &= \bigcup_{c\in\mathbb{N}} \mathsf{DSPACE}(n^c) = \bigcup_{c\in\mathbb{N}} \mathsf{NSPACE}(n^c)
\end{aligned}$$

For a class of languages $\mathcal{C}$, we denote by $\mathsf{co}\mathcal{C}$ the class of languages whose complements belong to $\mathcal{C}$. Note that our definitions of L, NL, P, NP and PSPACE coincide with the classical definitions of these complexity classes: random-access Turing machines can be simulated by classical multi-tape Turing machines with a factor n time overhead and no additional space.

Input encoding plays an important role when considering Turing machines with running time in $\mathcal{O}(\log^c n)$. It was shown in [BIS90] that several simple computations can be performed by random-access Turing machine in logarithmic time, including

- addition and subtraction of two $\mathcal{O}(\log n)$ bit numbers,
- computation of the logarithm of a $\mathcal{O}(\log n)$ bit number, and
- determining the length of the input.

One can also fetch words of length $\mathcal{O}(\log n)$ from a given address in logarithmic time since increments on the address tape can be realized in amortized constant time. However, it is not clear whether two $\mathcal{O}(\log n)$ bit numbers can be multiplied in logarithmic time. Therefore, whenever tables, matrices, lists or pairing functions are used to encode complex inputs, we require the encoding to be performed in a way that allows for easily accessing individual elements. For example, we can align the elements of a $n \times n$ bit matrix such that the entry in row i and column j is found at position $i \cdot 2^{\lceil \log n \rceil} + j$. This is similar to the usual *row-major* order but the alignment at powers of 2 guarantees that the mapping of a row and a column to an input position is $\mathsf{DLOGTIME}$-computable. All positions that are not in the image of the pairing function can be padded by a special character.

The complexity classes $\mathsf{DTIME}(\log^c n)$ (resp. $\mathsf{NTIME}(\log^c n)$) form a strict hierarchy within $\mathsf{DPOLYLOGTIME}$ (resp. $\mathsf{NPOLYLOGTIME}$): deciding whether the first $\log^{c+1} n$ bits of a given bit string of length n are 1 is easily seen to be in $\mathsf{DTIME}(\log^{c+1} n)$ but not in $\mathsf{NTIME}(\log^c n)$. Also note that $\mathsf{NLOGTIME}$ is not contained in $\mathsf{DPOLYLOGTIME}$: with a non-deterministic random-access Turing machine, it is possible to check in logarithmic time whether the input contains at least one 1, whereas linear time is required to decide this property on a deterministic random-access Turing machine.

2.4.2 Alternating Random-Access Turing Machines

In analogy to the definitions in the previous subsection, we can define *alternating random-access Turing machines*. Such a machine is defined like a non-deterministic random-access Turing machine but additionally has a *type* assigned to each state. The two possible types are *existential* and *universal*. By extension, we say that a configuration is existential (resp. universal) if it is in an existential (resp. universal) state. A configuration of such a machine is *accepting* if one of the following properties holds:

- The configuration is existential and there exists a (direct) successor configuration which is accepting.
- The configuration is universal and all (direct) successor configurations are accepting.

In particular, a universal configuration without successors always accepts and an existential configuration without successors always rejects. The machine is said to *run in time $t \in \mathbb{N}$ and with $k \in \mathbb{N}$ alternations on a configuration C* if the (full) computation tree with root C has height at most t and along every path (from the root to a leaf), there are at most k switches between existential and universal states. Note that if the number of alternations is zero, the configuration is accepting if and only if it is universal.

Acceptance and resource bounds are then defined as usual: an alternating random-access Turing machine *accepts a language L* if the start configuration associated with a word w is accepting if and only if $w \in L$. Let $t\colon \mathbb{N} \to \mathbb{N}$ be a function and let $k \in \mathbb{N}$. An alternating random-access Turing machine *runs in time t and with k alternations* if it runs in time $t(n)$ and with k alternations on the initial configuration of every input of length n.

For a function $f\colon \mathbb{N} \to \mathbb{N}$ and a number $k \in \mathbb{N}$, we write $\Sigma_{\mathsf{k}}\mathsf{TIME}(f(n))$ to denote the class of all languages accepted by an alternating random-access Turing machine which has a single existential initial state, runs in time $\mathcal{O}(f(n))$ and with k alternations. The definition of $\Pi_{\mathsf{k}}\mathsf{TIME}(f(n))$ is analogous with the initial state being universal instead of existential. It is easy to see that $\Pi_{\mathsf{k}}\mathsf{TIME}(f(n)) = \mathsf{co}\Sigma_{\mathsf{k}}\mathsf{TIME}(f(n))$.

The *logarithmic time hierarchy* LH is defined as $\bigcup_k \Sigma_\mathsf{k}\mathsf{TIME}(\log n)$. Analogously, the *polylogarithmic time hierarchy* PolyLH is defined as $\bigcup_k \Sigma_\mathsf{k}\mathsf{TIME}(\log^k n)$. The class LH is a strict superset of $\mathsf{NLOGTIME}$ and PolyLH is a strict superset of $\mathsf{NPOLYLOGTIME}$. To see this, note that the problem of deciding whether the input is of the form 1^n is in $\mathsf{coNTIME}(\log n)$ but not even in $\mathsf{NPOLYLOGTIME}$. Both LH and PolyLH have close connections to circuit complexity classes as described in the next section.

2.4.3 Circuit Complexity

A function has *quasi-polynomial* growth if it belongs to $2^{\mathcal{O}(\log^k n)}$ for some fixed $k \in \mathbb{N}$. Throughout the paper, we denote by AC^0 (resp. qAC^0) the class of languages decidable by unbounded fan-in Boolean circuit families of polynomial size (resp. quasi-polynomial size) and constant depth. We allow NOT gates but do not count them when measuring the depth or the size of a circuit. We will also talk about functions computed by these circuits families. The complexity classes ACC^0, TC^0, NC^1 and NC^2 are briefly referred to but their definitions are not needed.

Whenever circuit families are discussed, one needs to address the issue of *uniformity*. The definitions above correspond to the *non-uniform* variants of circuit complexity classes. A family of circuits is $\mathsf{DLOGTIME}$-*uniform* if its *direct connection language* is in $\mathsf{DLOGTIME}$; see [BIS90, Vol99] for details. A language is in $\mathsf{DLOGTIME}$-*uniform* AC^0 if it is decidable by a $\mathsf{DLOGTIME}$-uniform unbounded fan-in Boolean circuit family of polynomial size and constant depth. We denote by FOLL the class of languages decidable by $\mathsf{DLOGTIME}$-uniform unbounded fan-in Boolean circuit families of polynomial size and depth in $\mathcal{O}(\log \log n)$.

The Parity language is defined as $\{w \in \{0,1\}^* \mid |w|_1 \text{ is even}\}$. We often identify this language with its characteristic function and talk about the Parity function instead. It is known that this function cannot be computed by AC^0, FOLL or qAC^0 circuits. This follows directly from Håstad's and Yao's famous lower bound results [Has86, Yao85], which state that the number of Boolean gates required for a depth-d circuit to compute Parity is exponential in $n^{1/(d-1)}$.

It is known that LH equals $\mathsf{DLOGTIME}$-uniform AC^0; see e.g. [BIS90]. We will often use this correspondence to show that a function can be computed by a $\mathsf{DLOGTIME}$-uniform family of AC^0 circuits by showing instead that we can construct an alternating logarithmic-time random-access Turing machine with bounded alternations which computes, on a given input and an additional input number i, the i-th bit of the function. We also know that that $\mathsf{NTIME}(\log^2 n)$ is not a subset of AC^0: given the adjacency matrix of a graph and two vertices s and t, a non-deterministic random-access Turing machine can check in time $\mathcal{O}(\log^2 n)$ whether there exists an s-t-path of length $\log n$ whereas constant-depth circuits deciding this problem must have super-polynomial size [BIP98, COST16]. Conversely, since we already saw that LH is not contained in $\mathsf{NPOLYLOGTIME}$, we also know that AC^0 is not contained in $\mathsf{NPOLYLOGTIME}$. However, one can prove $\mathsf{PolyLH} \subseteq \mathsf{qAC}^0$ using a similar technique as for $\mathsf{LH} \subseteq \mathsf{AC}^0$. For completeness, we sketch the proof below.

Theorem 2.13. $\mathsf{PolyLH} \subseteq \mathsf{qAC}^0$.

Proof. Let $L \in \mathsf{PolyLH}$ be some language and let M be a fixed alternating random-access Turing machine which runs in time $\mathcal{O}(\log^k n)$ and with k alternations. We will construct a circuit for a fixed input size n so the time bound $t(n) \in \mathcal{O}(\log^k n)$ is fixed.

Let us make a preliminary consideration. A path from a configuration C to a configuration C' in the computation tree is called *alternation-free* if every inner configuration on that path has the same type as C — we allow that C' has a different type. For every fixed pair of configurations (C, C') we can test whether, in the computation tree, there exists an alternation-free path of length at most $t(n)$ from C to C'. To this end, note that the number of different configurations is in $2^{\mathcal{O}(\log^k n)}$ and there are only $2^{\mathcal{O}(\log^{2k} n)}$ sequences of configurations of length at most $t(n)$. For each such sequence, we can easily check, using a constant-depth polylogarithmic-size circuit, whether the sequence corresponds to a valid alternation-free path in the computation tree. We do this for all sequences in parallel. The result is then obtained using a single additional OR gate. We precompute this reachability predicate for all pairs of configurations.

Now, for every $i \in \{0, \dots, k\}$, we construct a quasi-polynomial-size constant-depth circuit which decides, for a fixed configuration C, whether M runs in i alternations on C and if so, whether or not C is accepting. The proof is by induction on i and the circuit for i reuses values computed by the circuit for $i-1$. For $i = 0$, we simply check whether no configuration of a different type than C is reachable from C by an alternation-free path — this can be implemented with a single OR gate which is connected to gates of the reachability precomputation layer. We then mark the configuration as accepting if and only if C is universal.

Let $i \geqslant 1$. By induction, we know that for each fixed configuration, there exists a quasi-polynomial-size constant-depth circuit deciding whether the machine runs in $i-1$ alternations on that configuration and if so, whether the configuration is accepting. Since there are only $2^{\mathcal{O}(\log^k n)}$ possible configurations of M, we can think of having precomputed these predicates for all configurations.

Now, to check whether M runs with i alternations on C, we check, for all configurations C' reachable by an alternation-free path from C and of different type than C, whether M runs with $i-1$ alternations on C'. Assume now that M indeed runs with i alternations on C. If C is existential, we check that there exists a universal configuration C' which is both reachable from C by an alternation-free path and accepting. Again, this is done by considering all possible C' in parallel. Note that by assumption, the machine runs in less than i alternations on each such C'. Similarly, if C is universal, we check that all existential configurations C' reachable by an alternation-free path are accepting. □

We remark that proving membership to the polylogarithmic time hierarchy does not immediately yield efficient deterministic sequential algorithms: the best known upper bound on the deterministic time complexity of problems in PolyLH is quasi-polynomial. Most of the problems, which are considered in this work and fall into the polylogarithmic time hierarchy, have been previously known to be decidable in deterministic polynomial time. Nevertheless, the existence of PolyLH algorithms has strong consequences. Together with Håstad's and Yao's lower bound results, the previous theorem implies that PolyLH does not contain Parity. Therefore, problems in PolyLH cannot be hard for many natural complexity classes such as ACC^0, TC^0, NC^1, L or NL (at least under non-uniform qAC^0 reductions). Different variants of reducibility, hardness and completeness are introduced in the next part of this section on complexity.

2.4.4 Reductions and Hardness Results

A language K is *reducible* to a language L *via* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions* if L is decidable by a $\mathsf{DLOGTIME}$-uniform family of unbounded fan-in constant-depth Boolean circuits with oracle gates for K. In this case, we write $K \leqslant_{\mathsf{AC}^0} L$. If both $K \leqslant_{\mathsf{AC}^0} L$ and $L \leqslant_{\mathsf{AC}^0} K$, we also write $K \equiv_{\mathsf{AC}^0} L$ and say that K and L are *equivalent under* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions*.

Let $\mathcal{C}$ be a complexity class. A problem L is said to be $\mathcal{C}$*-hard under* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions* if every language in $\mathcal{C}$ is reducible to L via $\mathsf{DLOGTIME}$-uniform AC^0 reductions. If, additionally, $L \in \mathcal{C}$, then L is $\mathcal{C}$*-complete under* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions*. For other circuit complexity classes, such as non-uniform AC^0 or qAC^0, the notions of reductions, equivalence hardness and completeness can be defined in a similar manner. Many reducibility and completeness results in this thesis also hold for more stringent notions of reducibility, such as *many-one reductions*.

We will also use many-one reductions for log-space reducibility. Formally, a language K is *reducible* to a language L *via log-space reductions* if there exists a log-space computable function f such that $w \in K$ if and only if $f(w) \in L$. Recall that a function is called log-space computable if there exists a sequential Turing machine with a work tape of logarithmic size and a write-only, write-once output tape. Hardness and completeness under log-space reductions are defined as for $\mathsf{DLOGTIME}$-uniform AC^0 reductions. All hardness and completeness results in this thesis are stated in terms of log-space reducibility, unless stated otherwise.

Reachability in directed graphs is NL-complete under $\mathsf{DLOGTIME}$-uniform AC^0 reductions and reachability in undirected graphs is L-complete under $\mathsf{DLOGTIME}$-uniform AC^0 reductions [Rei08]. The satisfiability problem for formulas conjunctive normal form with at most three literals per clause, denoted by 3-SAT, is NP-complete.

2.4.5 Encoding

Encoding plays an important role when considering "low" complexity classes such as $\mathsf{DLOGTIME}$ or AC^0. In most problems considered in this thesis, the input consists of a semigroup S (or multiple semigroups). Sometimes, we are additionally given a set of generators of a subsemigroup of S, a morphism $h \colon A^+ \to S$ or an accepting set $P \subseteq S$. We consider two different encodings of finite semigroups.

In most cases, finite semigroups are given by their multiplication table, often also called *Cayley table*. Following this terminology, we call such an encoding *Cayley encoding*. Given a finite semigroup of cardinality n, its elements are identified with the set of integers $\{1, \ldots, n\}$, such that the length of the encoding of a single element is in $\mathcal{O}(\log n)$. The full multiplication table is given as a matrix in row-major order and we often assume that all entries and rows are padded such that the position of the element in a given row i and column j is $\mathsf{DLOGTIME}$ computable; see Section 2.4.1 for details. We also do not require that every entry of the table corresponds to a semigroup element: the Cayley table may contain gaps, i.e., rows and columns which are filled with a distinguished *padding value*. This relaxation allows to use, say, group algorithms on subgroups of a semigroup without having to take care of rearranging the entries of the Cayley table. More generally, given the Cayley encoding of a semigroup S and a set of elements forming subsemigroup T of S, we can easily transform the Cayley table of S to a Cayley encoding of T by overwriting all rows and columns not in T with the padding value.

We assume that the dimensions of the Cayley table are known, which gives a sufficiently good upper bound on the size of the semigroup. To this end, we can either add length descriptors to the input or use linear search to obtain the length of a table entry, then use a double binary search to obtain the length of the first row.

Sets of generators are given as a list of integers corresponding to the elements of the semigroup, unless stated otherwise. We may assume that the list entries are aligned such that their addresses are $\mathsf{DLOGTIME}$-computable. Moreover, we assume that either a length descriptor is provided or the list is padded with special values to a large enough power of 2 such that its length can be determined by a double binary search in deterministic logarithmic time. When a morphism $h \colon A^+ \to S$ is given, we think of having an implicit strict linear order on the letters $a_1, \ldots, a_m$ of A and the morphism is given as a list of elements $h(a_1), \ldots, h(a_m)$ with $a_1 < \cdots < a_m$. Essentially, the morphism is encoded in the same way as we would encode the set of generators $\{h(a) \mid a \in A\}$ but the order on A is respected. The order is particularly important when multiple morphisms from a common alphabet are provided in the input in which case we assume a common ordering. Accepting sets $P \subseteq S$ are usually represented as bit vectors instead. This allows a random-access Turing machine to verify in deterministic logarithmic time whether or not an element belongs to P. Providing accepting sets as a list requires performing a linear search instead.

The second encoding we consider is for *transformation semigroups*. Here, we choose a representation which closely resembles the classical DFA representation. The semigroup is always represented by a set of generators. Each generator is encoded as a list $f(1), \ldots, f(n)$ representing a function $f \colon Q \to Q$ on some common set $Q = \{1, \ldots, n\}$. As in the case of Cayley encodings, if a morphism is given, it is represented as the ordered list of corresponding generators with the order induced by an implicit order on the alphabet. Note that the individual elements of the semigroup are *not* part of the input and are given implicitly as a subsemigroup of the full transformation semigroup on Q. In particular, the size of the semigroup can be exponential in the input size. For an accepting set, to uphold the correspondence with DFAs, we specify a distinguished element $q_0 \in Q$ and a set $F \subseteq Q$ to represent the set of all transformations $\{f \colon Q \to Q \mid f(q_0) \in F\}$. The element q_0 is represented as an integer from the set $\{1, \ldots, n\}$ and F is represented as a bit vector.

The following lemma revisits the conversion of semigroups to deterministic finite automata and shows that the conversion can be carried out efficiently with the encodings described above.

Lemma 2.14. *Let S be a finite semigroup and let X be a subset of S, generating a subsemigroup T of S. Then, there exist a set Q and a set Y of transformations on Q such that the transformation semigroup generated by Y is isomorphic to T. Moreover, given the Cayley encoding of S and the set X (as a list of integers representing elements of S), the set Y can be computed by a* $\mathsf{DLOGTIME}$*-uniform family of* AC^0 *circuits.*

Proof. As described in Section 2.3, it suffices to construct the encoding of the transformation semigroup on S^1 generated by the transformations $Y = \{f_x \colon S^1 \to S^1 \mid x \in X\}$ where $f_x(s) = s \cdot x$ for $s \in S^1$ and for $x \in X$. We will actually construct a transformation semigroup on a bigger set Q. This set consists of all rows of the Cayley table of S, including rows filled with the padding value. The idea is that all padding rows are mapped to a distinguished new element of Q in each of the transformations. This yields a subsemigroup of the full transformation semigroup which is isomorphic to the semigroup obtained by restricting all transformations to the non-padding elements. The latter semigroup is, in turn, isomorphic to S.

We let $Q = \{1, \dots, n+2\}$ where n is the number of rows in the Cayley encoding of S. The value $n+1$ is used for the neutral element added to S and the value $n+2$ is used for the special padding value. Now, for each $x \in X$, the vector corresponding to f_x is constructed by copying a column of the Cayley table and replacing every entry with the padding value by $n+2$. The second last entry of the vector is filled with x itself and the last entry is filled with the value $n+2$.

To see that the computation can be performed by a family of $\mathsf{DLOGTIME}$-uniform AC^0 circuits, it suffices to show that a variant of the computation can be carried out in $\mathsf{DLOGTIME}$. In this variant, arguments $i \in \{1, \dots, |X|\}$ and $j \in \{1, \dots, n+2\}$ are added to the input, and we are only supposed to compute the j-th entry of the i-th transformation. For $j \leqslant n$, this is achieved by fetching the i-th element x_i of the list representing X first and then performing a table lookup to obtain the entry in row x_i and column j of the multiplication table. The padding value is replaced by $n+2$. For $j = n+1$, we simply copy the value x_i instead. For $j = n+2$, we return $n+2$. □

When multiple semigroups are given in the input, we again assume a suitable encoding, i.e., the semigroups, morphisms and accepting sets are all padded to a common power of 2 and aligned in a way such that given an integer i, the addresses of the i-th Cayley table, the i-th morphism and the i-th accepting set are $\mathsf{DLOGTIME}$-computable.

Chapter 3

Building Blocks for Efficient Algorithms

3.1 First-Order Definable Properties

In this section, we describe how to express properties of finite semigroups using first-order logic. The basic idea is straightforward. For example, the property of having a neutral element can be expressed as

$$\exists e \forall x \colon (ex = x \,\wedge\, xe = x).$$

Although we are mostly interested in applying the results and techniques of this section to finite semigroups, some results are stated in the more general setting of *partial semigroups*, i.e., non-empty sets with an associative partial binary operation. Note that associativity is also required in the case that the result of a multiplication is undefined, i.e., $(xy)z$ is undefined if and only if $x(yz)$ is undefined. The main reason for considering partial semigroups is that for certain closure operations, we want to apply formulas to subsets of finite semigroups. These subsets need not form subsemigroups but can be interpreted as partial semigroups in a natural way with the result of a multiplication being undefined whenever the element does not belong to the considered subset. We denote the class of all finite partial semigroups by **PS**.

As usual, first-order formulas are built upon the logical connectives $\vee$, $\neg$, the quantifier $\exists$, a set of variables X, the equality symbol and elements from a set of further symbols which is called signature. This *signature* τ is a set consisting of *relation symbols* (also called *predicates*) and *function symbols*. Each relation symbol and each function symbol has fixed arity. The set of all *τ-terms* is the least set containing all variables X and closed under building expressions $f(t_1, \dots, t_m)$ where f is a function symbol of arity m from τ and $t_1, \dots, t_m$ are themselves τ-terms. For a signature τ, the expressions of the form $t_1 = t_2$ or $R(t_1, \dots, t_m)$, where $t_1, \dots, t_m$ are τ-terms and R is an m-ary relation symbol from τ, are called *atomic τ-formulas*. The set of FO[τ]*-formulas* is then defined inductively: each atomic τ-formula is an FO[τ]-formula and if φ and ψ are FO[τ]-formulas and x is a variable, then $\varphi \vee \psi$, $\neg\varphi$ and $\exists x \colon \varphi$ are FO[τ]-formulas as well. We will use common abbreviations such as $\forall x \colon \varphi$ instead of $\neg\exists x \colon \neg\varphi$, the abbreviation $\varphi \to \psi$ for $\neg\varphi \vee \psi$ and $\varphi \leftrightarrow \psi$ instead of $(\varphi \wedge \psi) \vee (\neg\varphi \wedge \neg\psi)$. A variable is said to occur *freely* in an FO[τ]-formula if it appears outside the scope of a quantifier.

Sometimes, we write $\varphi(x_1, \ldots, x_m)$ to indicate that at most the variables $x_1, \ldots, x_m$ occur freely in φ. A *sentence* is a formula without any free occurrences of variables.

As mentioned before, we consider $\mathrm{FO}[\tau]$-formulas over finite partial semigroups, i.e., signatures where every finite partial semigroup uniquely defines the semantics of all symbols in τ. Variables then correspond to elements of a finite partial semigroup S. Thus, one can evaluate every symbol of a signature τ for a given finite partial semigroup S and an assignment of elements of S to its parameters without providing further information on its semantics. The *truth value* of an $\mathrm{FO}[\tau]$-formula $\varphi(x_1, \ldots, x_m)$ in a partial semigroup S for an *assignment* of elements $s_1, \ldots, s_m \in S$ to the variables $x_1, \ldots, x_m$ is defined inductively as usual, and we write $(S, s_1, \ldots, s_m) \models \varphi(x_1, \ldots, x_m)$ if the assignment specified by S and by assigning each s_i to the corresponding x_i satisfies φ. Tuples of the form $(S, s_1, \ldots, s_m)$ are called *structures*. Two $\mathrm{FO}[\tau]$-formulas are *equivalent* if they are satisfied by exactly the same structures.

The relation symbol $\cdot$ is interpreted as the binary operation in the corresponding finite partial semigroup S: it consists of all tuples (x, y, z) such that $xy = z$ in S. When using the symbol in formulas, we usually write $xy = z$ instead of $\cdot(x, y, z)$. Note that $xy = z$ does not hold whenever the product xy is undefined.

A useful property of first-order formulas is that computing the truth value reduces to computing the truth value of the predicates via weak reductions.

Theorem 3.1 (Immerman)**.** *Let $\varphi(x_1, \ldots, x_m)$ be an $\mathrm{FO}[\tau]$-formula. Then, the problem of deciding whether, for a given finite partial semigroup S in Cayley encoding and given elements $s_1, \ldots, s_m \in S$, the relation $(S, s_1, \ldots, s_m) \models \varphi$ holds is reducible to computing the predicates and functions in τ via $\mathsf{DLOGTIME}$-uniform AC^0 reductions.*

We are mainly interested in first-order formulas with multiplication. For details on the generic case, we refer to [BIS90]. As a direct consequence of the theorem above, one can compute the truth value of first-order formulas with multiplication in $\mathsf{DLOGTIME}$-uniform AC^0. We sharpen this result by considering the *alternation depth* of formulas.

Allowing the use of universal quantifiers, we can use De Morgan's laws to convert any $\mathrm{FO}[\tau]$-formula φ to an equivalent formula where negations only appear in front of atomic formulas. This formula is called the *negation-free counterpart* of φ. The alternation depth of φ is the maximum number of blocks of existential and universal quantifiers along each root-to-leaf path in the syntax tree of the negation-free counterpart of φ. For $k \in \mathbb{N}$, the set of $\mathrm{FO}_k[\tau]$-formulas is the set of all $\mathrm{FO}[\tau]$-formulas with alternation depth at most k. The set of $\Sigma_k[\tau]$-formulas (resp. $\Pi_k[\tau]$-formulas) is obtained by additionally requiring every path with k blocks of quantifiers to start with an existential (resp. universal) quantifier. Equivalently, for $k \geqslant 1$, the formula φ is a $\Sigma_k[\tau]$-formula (resp. $\Pi_k[\tau]$-formula) whenever $\exists x\colon \varphi$ (resp. $\forall x\colon \varphi$) is an $\mathrm{FO}_k[\tau]$-formula.

Theorem 3.2. *Let $k \geqslant 1$ and let $\varphi(x_1, \ldots, x_m)$ be a $\Sigma_k[\,\cdot\,]$-formula. Then, the problem of deciding whether a given finite partial semigroup S in Cayley encoding and given elements $s_1, \ldots, s_m \in S$ satisfy $(S, s_1, \ldots, s_m) \models \varphi$ belongs to $\Sigma_\mathsf{k}\mathsf{TIME}(\log n)$.*

Proof. We can convert the negation-free counterpart of φ into prenex normal form (i.e., a formula consisting of a prefix of quantifiers, followed by a quantifier-free part) without affecting the alternation depth. In the resulting formula φ', there are still at most $k-1$ alternations between existential and universal quantifiers, and if the number of alternations is exactly $k-1$, the first quantifier block is existential.

The construction of an alternating random-access Turing machine for φ' is straightforward. The machine evaluates the formula top-down. For every block of existential quantifiers, we branch existentially and guess an assignment of elements to all variables occurring in the block. For every block of universal quantifiers, we use universal states to explore all possible assignments to the quantified variables. The verification of the quantifier-free part of the formula is purely sequential. For atomic subformulas, we perform a single table lookup to check whether or not the atomic formula is satisfied under the guessed assignment. The table lookup requires $\mathcal{O}(\log n)$ time assuming a suitable encoding of the multiplication table of S as described in Section 2.4.5.

Clearly, the resulting machine runs in logarithmic time since the formula is fixed and every guessing step, as well as every verification of an atomic formula requires logarithmic time. Moreover, the machine can be designed to start in an existential state and only perform k alternations on every computation path: the first $k-1$ alternations are required for quantifier alternations in the formula and the last alternation is required for either accepting or rejecting the quantifier-free subformula. □

The same construction yields $\mathsf{DTIME}(\log n)$ algorithms for $\mathrm{FO}_0[\,\cdot\,]$-formulas. Formulas with free variables as in the statement of the previous theorem can be viewed as a description of families of relations of fixed arity. Formally, a *relation on partial semigroups* of arity m is a family of relations $(R_S)_{S\in\mathbf{PS}}$ such that $R_S \subseteq S^m$ for each partial semigroup S. For a given relation on partial semigroups R and a finite partial semigroup S, we use R_S to denote the relation in R associated to S. A relation on partial semigroups R of arity m is *defined by* an $\mathrm{FO}[\tau]$-formula $\psi(x_1,\dots,x_m)$ if for all finite partial semigroups S, we have $(s_1,\dots,s_m)\in R_S$ if and only if $(S,s_1,\dots,s_m)\models\psi(x_1,\dots,x_m)$. For an $\mathrm{FO}[\tau]$-sentence φ, the class $\mathbf{C}$ of finite partial semigroups *defined by* φ is the class of finite partial semigroups S such that $S\in\mathbf{C}$ if and only if $S\models\varphi$. The notions of $\mathrm{FO}_k[\tau]$-, $\Sigma_k[\tau]$-, and $\Pi_k[\tau]$-*definability* are defined in a straightforward manner.

First-order formulas with multiplication are powerful enough to express a variety of relations commonly occurring in the structure theory of finite semigroups.

Lemma 3.3. *Green's relations $\leqslant_{\mathcal{R}}$, $\leqslant_{\mathcal{L}}$, $\leqslant_{\mathcal{J}}$, $\leqslant_{\mathcal{H}}$, $\mathcal{R}$, $\mathcal{L}$, $\mathcal{J}$ and $\mathcal{H}$ are $\Sigma_1[\,\cdot\,]$-definable.*

Proof. We have $x \leqslant_{\mathcal{R}} y$ if and only if $x = y \vee \exists z\colon yz = x$ and $x \mathrel{\mathcal{R}} y$ if and only if $x \leqslant_{\mathcal{R}} y$ and $y \leqslant_{\mathcal{R}} x$. A similar construction can be used for the other relations. □

Lemma 3.4. *Given the Cayley table of a finite semigroup S and two elements $s,t\in S$, the problem of deciding whether $s \leqslant_{\mathcal{R}} t$ (resp. $s \leqslant_{\mathcal{L}} t$, $s \leqslant_{\mathcal{J}} t$, $s \leqslant_{\mathcal{H}} t$, $s \mathrel{\mathcal{R}} t$, $s \mathrel{\mathcal{L}} t$, $s \mathrel{\mathcal{J}} t$ or $s \mathrel{\mathcal{H}} t$) is in* $\mathsf{NLOGTIME}$.

Proof. This follows immediately from Lemma 3.3 and Theorem 3.2. □

We will make use of this observation and henceforth, Green's relations will be used as abbreviations in some $\mathrm{FO}[\,\cdot\,]$-formulas without further explanation. The next lemma shows how Green's relations can be used to express the ω-operator.

Lemma 3.5. *Let S be a finite semigroup and let $s \in S$. Then s^ω is the unique $\leqslant_{\mathcal{H}}$-maximal idempotent element $t \in S$ such that $st \mathrel{\mathcal{R}} t$ and $ts \mathrel{\mathcal{L}} t$.*

Proof. Since $\mathcal{R}$ is stable on the left, every element $t \in S$ with $st \mathrel{\mathcal{R}} t$ satisfies $s^{i+1}t \mathrel{\mathcal{R}} s^i t$ for all $i \geqslant 0$. By transitivity, we obtain $s^i t \mathrel{\mathcal{R}} t$ for all $i \geqslant 0$. In particular, $s^\omega t \mathrel{\mathcal{R}} t$, and thus, $t \leqslant_{\mathcal{R}} s^\omega$. Symmetrically, $ts \mathrel{\mathcal{L}} t$ yields $t \leqslant_{\mathcal{L}} s^\omega$. Therefore, we have $t \leqslant_{\mathcal{H}} s^\omega$ for every element $t \in S$ with $st \mathrel{\mathcal{R}} t$ and $ts \mathrel{\mathcal{L}} t$.

Now, since $s^{\omega+1} \mathrel{\mathcal{H}} s^\omega$, every $\leqslant_{\mathcal{H}}$-maximal element $t \in S$ with $st \mathrel{\mathcal{R}} t$ and $ts \mathrel{\mathcal{L}} t$ satisfies $t \mathrel{\mathcal{H}} s^\omega$. Since every $\mathcal{H}$-class contains at most one idempotent element, if t is idempotent, we have $t = s^\omega$, as desired. □

Thus, the ω-operator can be added to first-order logic with multiplication without changing the expressive power. Formally, the predicate $\omega(x, y)$ contains all tuples (s, t) with $s^\omega = t$ in S. To improve readability, we usually write $x^\omega = y$ instead of $\omega(x, y)$.

Proposition 3.6. *Let τ be a set of predicates with the binary operation $\cdot$ contained in τ. Over semigroups, every $\mathrm{FO}_k[\tau \cup \{\omega\}]$-formula is equivalent to some $\mathrm{FO}_{k+1}[\tau]$-formula. Similarly, every $\Sigma_k[\tau \cup \{\omega\}]$-formula is equivalent to some $\Sigma_{k+1}[\tau]$-formula.*

The class of all finite semigroups $\mathbf{S}$ is defined by the $\mathrm{FO}[\,\cdot\,]$-formula $\forall x \forall y \exists z \colon xy = z$. Together with the previous proposition, this shows that every variety of finite semigroups defined by a finite set of ω-identities can be defined by an $\mathrm{FO}[\,\cdot\,]$-formula. However, the converse is not true: $\mathrm{FO}[\,\cdot\,]$-formulas are strictly more expressive than finite sets of ω-identities, even when restricted to varieties of finite semigroups. For example, the variety of finite solvable groups is known to be not even definable using infinitely many ω-identities but can be described using a first-order formula [Wil06].

Theorem 3.7 (Wilson). *The variety of finite solvable groups $\mathbf{G}_{\mathrm{sol}}$ is $\mathrm{FO}[\,\cdot\,]$-definable.*

More examples of $\mathrm{FO}[\,\cdot\,]$-definable relations were given at the end of Section 2.2.2. In fact, the definitions of the relations RM, LM, GGM and AGGM already implicitly used the concept of first-order formulas over semigroups.

Lemma 3.8. *The relations RM, LM and GGM are $\Pi_1[\,\cdot\,]$-definable. The relation AGGM is $\Pi_2[\,\cdot\,]$-definable.*

We conclude this section by proving some closure properties of first-order definable classes of finite semigroups. It is clear that Boolean operations preserve first-order definability. On the other hand, it seems impossible to obtain very general results for other generic operations such as joins, Mal'cev products or *semidirect products*: it follows from [Rho99] that none of these operations preserve $\mathrm{FO}[\,\cdot\,]$-definability. Nevertheless, one can prove closure under *specific* Mal'cev products and other operations. These closure properties are based on the next two lemmas.

For a partial semigroup S, a binary relation $\sim$ on S and an element $s \in S$, one can restrict the multiplication in S to the set $[s]_\sim := \{t \in S \mid s \sim t\}$. A product is undefined whenever the corresponding product in S does not belong to $[s]_\sim$. Note that we use the notation $[s]_\sim$ for arbitrary relations (which are not necessarily equivalence relations). We are primarily interested in the case that the sets $[s]_\sim$ form partial semigroups.

Lemma 3.9. *Let* $\mathbf{C}$ *be an* $\mathrm{FO}_k[\,\cdot\,]$*-definable class of finite partial semigroups and let* $\sim$ *be an* $\mathrm{FO}_\ell[\,\cdot\,]$*-definable relation on partial semigroups. Then, the class of all finite partial semigroups* S*, such that sets* $[s]_{\sim_S}$ *form partial semigroups from* $\mathbf{C}$*, is* $\Pi_{k+\ell+1}[\,\cdot\,]$*-definable.*

Proof. Suppose that φ is an $\mathrm{FO}_k[\,\cdot\,]$-sentence with $S \models \varphi$ if and only if $S \in \mathbf{C}$ and suppose that $\psi(x, y)$ is a $\mathrm{FO}_\ell[\,\cdot\,]$-formula such that for all finite partial semigroups S and for all $s, t \in S$, we have $s \sim_S t$ if and only if $(S, s, t) \models \psi(x, y)$. We construct a $\Pi_{k+\ell+1}[\,\cdot\,]$-sentence φ'' such that for all finite partial semigroups S, we have $S \models \varphi''$ whenever all partial semigroups $[s]_{\sim_S}$ belong to $\mathbf{C}$, i.e., $[s]_{\sim_S} \models \varphi$.

Let x be a variable not appearing in φ. We successively replace all subformulas of the form $\exists y\colon \chi$ in φ by $\exists y\colon (\psi(x, y) \wedge \chi)$ to obtain a formula φ'. Note that x occurs freely in φ'. We then let $\varphi'' = \forall x\colon \varphi'(x)$.

In order to prove correctness of the construction, we prove a slightly stronger statement: suppose that the original formula φ contained additional free variables $z_1, \ldots, z_m$. We show that then, for all finite partial semigroups S, for all $s \in S$ and for all elements $r_1, \ldots, r_m \in [s]_{\sim_S}$, the condition $(S, s, r_1, \ldots, r_m) \models \varphi'(x, z_1, \ldots, z_m)$ is equivalent to $([s]_{\sim_S}, r_1, \ldots, r_m) \models \varphi(z_1, \ldots, z_m)$.

If φ is an atomic formula, the claim clearly holds. If φ has the form $\chi_1 \vee \chi_2$ or $\neg\chi$, the statement holds by induction. Suppose now that $\varphi(z_1, \ldots, z_m) = \exists y\colon \chi(y, z_1, \ldots, z_m)$ for some $\mathrm{FO}[\,\cdot\,]$-formula χ. Then, $\varphi'(x, z_1, \ldots, z_m) = \exists y\colon (\psi(x, y) \wedge \chi'(x, y, z_1, \ldots, z_m))$ where, by induction, we have that $(S, s, t, r_1, \ldots, r_m) \models \chi'(x, y, z_1, \ldots, z_m)$ if and only if $([s]_{\sim_S}, t, r_1, \ldots, r_m) \models \chi(y, z_1, \ldots, z_m)$. Using this equivalence and the definition of φ', we know that $(S, s, r_1, \ldots, r_m) \models \varphi'(x, z_1, \ldots, z_m)$ if and only if there exists some $t \in S$ with $s \sim_S t$ such that $([s]_{\sim_S}, t, r_1, \ldots, r_m) \models \chi(y, z_1, \ldots, z_m)$. By the definition of φ, the latter property is equivalent to $([s]_{\sim_S}, r_1, \ldots, r_m) \models \varphi(z_1, \ldots, z_m)$, as desired. □

A binary relation on partial semigroups $\sim$ is a *semigroup congruence on partial semigroups* if for each finite semigroup S, the relation $\sim_S$ is a congruence.

Lemma 3.10. *Let* $\mathbf{C}$ *be an* $\mathrm{FO}_k[\,\cdot\,]$*-definable class of finite semigroups and let* $\sim$ *be an* $\mathrm{FO}_\ell[\,\cdot\,]$*-definable semigroup congruence on partial semigroups. Then the class of all finite semigroups* S *with* $S/{\sim_S} \in \mathbf{C}$ *is* $\mathrm{FO}_{k+\ell}[\,\cdot\,]$*-definable.*

Proof. Suppose that φ is an $\mathrm{FO}_k[\,\cdot\,]$-sentence with $S \models \varphi$ if and only if $S \in \mathbf{C}$. Suppose $\psi(x, y)$ is an $\mathrm{FO}_\ell[\,\cdot\,]$-formula such that for all finite semigroups S and for all $s, t \in S$, we have $s \sim_S t$ if and only if $(S, s, t) \models \psi(x, y)$. We construct an $\mathrm{FO}_{k+\ell}[\,\cdot\,]$-sentence φ' such that for every finite semigroup S, we have $S \models \varphi'$ if and only if $S/{\sim_S} \models \varphi$.

This sentence φ' is obtained by starting with the negation-free counterpart of φ and successively replacing all positively occurring atomic formulas of the form $xy = z$ by the formula $\exists w\colon (xy = w \wedge \psi(w, z))$ whenever the closest quantifier to the occurrence of $xy = z$ in the syntax tree is existential. The same replacement is performed if, in the syntax tree, the parent of $xy = z$ is a negation and the closest quantifier is universal. All other atomic formulas are replaced by $\forall w\colon (xy = w \rightarrow \psi(w, z))$. Over semigroups, the two replacement formulas $\exists w\colon (xy = w \wedge \psi(w, z))$ and $\forall w\colon (xy = w \rightarrow \psi(w, z))$ are equivalent. In any case, w is a fresh variable not appearing in φ.

As in the proof of Lemma 3.9, we prove a stronger statement with additional free variables: for every finite semigroup S, the conditions $(S, s_1, \dots, s_m) \models \varphi'(z_1, \dots, z_m)$ and $(S/{\sim_S}, [s_1]_{\sim_S}, \dots, [s_m]_{\sim_S}) \models \varphi(z_1, \dots, z_m)$ are equivalent for all $s_1, \dots, s_m \in S$.

If φ is an atomic formula $xy = z$, then the quotient $S/{\sim_S}$ satisfies φ if and only if classes $[s]_{\sim_S}$, $[t]_{\sim_S}$ and $[u]_{\sim_S}$ are assigned to x, y, z such that $[s]_{\sim_S}[t]_{\sim_S} = [u]_{\sim_S}$, i.e., $st \sim_S u$. This is equivalent to the existence of some element $r \in S$ such that $st = r$ and $r \sim_S u$, which is exactly what $\varphi' = \exists w \colon (xy = w \wedge \psi(w, z))$ expresses. Induction yields the desired statement if φ has the form $\chi_1 \vee \chi_2$, $\neg\chi$ or $\exists y \colon \chi$. □

We obtain the following closure properties. Applications of first-order formulas over finite semigroups are given in Section 5.5.

Theorem 3.11. *Let* $\mathbf{V}$ *be an* $\mathrm{FO}_k[\,\cdot\,]$*-definable variety of finite semigroups and let* $\mathbf{H}$ *be an* $\mathrm{FO}_k[\,\cdot\,]$*-definable variety of finite groups. Then, the classes* $\mathbb{D}\mathbf{V}$, $\mathbb{L}\mathbf{V}$, $\mathbf{V}^{\mathbf{N}}$ *and* $\overline{\mathbf{H}}$ *are* $\Pi_{k+2}[\,\cdot\,]$*-definable. Moreover, the classes* $\mathbf{K} \circledcirc \mathbf{V}$, $\mathbf{D} \circledcirc \mathbf{V}$, $\mathbf{N} \circledcirc \mathbf{V}$, $\mathbb{L}\mathbf{I} \circledcirc \mathbf{V}$ *are* $\mathrm{FO}_{k+1}[\,\cdot\,]$*-definable and* $\mathbb{L}\mathbf{G} \circledcirc \mathbf{V}$ *is* $\mathrm{FO}_{k+2}[\,\cdot\,]$*-definable.*

Proof. To see that $\mathbb{D}\mathbf{V}$ is $\Pi_{k+2}[\,\cdot\,]$-definable, we define a relation on finite partial semigroups $\sim$ by $x \sim_S y$ if and only if $x^2 = x \wedge x \mathrel{\mathcal{J}} y$. Moreover, given a finite semigroup S, the non-empty sets $\{t \in S \mid s \sim_S t\}$ correspond to the regular $\mathcal{J}$-classes of S. Thus, $\mathbb{D}\mathbf{V}$ is $\Pi_{k+2}[\,\cdot\,]$-definable by Lemma 3.9. The same arguments can be used for $\mathbb{L}\mathbf{V}$ (with $x \sim_S y$ if $x^2 = x \,\wedge\, \exists z \colon xzx = y$), for $\mathbf{V}^{\mathbf{N}}$ (with $x \sim_S y$ if $\exists p \exists z \exists q \colon (z^2 = z \wedge pzq = y)$) and for $\overline{\mathbf{H}}$ (with $x \sim_S y$ if $x^2 = x \,\wedge\, x \mathrel{\mathcal{H}} y$).

For the remaining classes, we combine Lemma 3.8 and Lemma 3.10. □

3.2 Cayley Circuits and Straight-Line Programs

Straight-line compression is a key technique used in many of the upcoming complexity results. It is an essential ingredient in the *circuits properties framework* introduced in Chapter 4 which, in turn, plays a central role in the complexity of the emptiness, universality, inclusion, equivalence, finiteness and intersection non-emptiness problems. Usually, *straight-line programs* are defined as context-free grammars over some fixed algebraic structure, and can also be viewed as *algebraic circuits*. In our applications, we are only interested in straight-line programs over semigroups. However, these semigroups are often not fixed but given as part of the input. Therefore, we first introduce the more abstract concept of *Cayley circuits*. These circuits only capture the structure of a straight-line program but do not fix the algebraic structure nor the constants. The terminology originates from the fact that one can view such circuits as producing a uniquely defined output once the *Cayley table* of a semigroup and assignments to the inputs are provided. Formally, a *Cayley circuit* $\mathcal{C}$ is a tuple $(V_{\mathsf{in}}, V_{\mathsf{mul}}, <, \ell, r)$ where

- V_{in} and V_{mul} are pairwise disjoint finite sets,
- $<$ is a (strict) linear order on $V := V_{\mathsf{in}} \cup V_{\mathsf{mul}}$, and
- $\ell \colon V_{\mathsf{mul}} \to V$ and $r \colon V_{\mathsf{mul}} \to V$ are mappings with $\ell(v), r(v) < v$ for all $v \in V_{\mathsf{mul}}$.

Such a tuple can be interpreted as a directed acyclic multi-edge graph with topologically ordered vertices V, each of which has in-degree 0 or 2. The predecessors of each in-degree-2 vertex are ordered. The elements of V are called *gates*, the elements of V_{in} are called *input gates* and the elements of V_{mul} are called *multiplication gates.* In the graph representation, the input gates correspond to the vertices of in-degree 0 and the multiplication gates correspond to the gates of in-degree 2. The *input arity* of $\mathcal{C}$ is the number of input gates $|V_{\mathsf{in}}|$ and the *size* of $\mathcal{C}$ is the total number of gates $|V|$. For $i \in \{1, \ldots, |V|\}$, let A_i be the set of the i smallest gates of V with respect to $<$ and let $B_i = V \setminus A_i$. The *width* of $\mathcal{C}$ is defined as

$$\max_{1 \leqslant i < |V|} \left|\left\{(\ell^{-1}(B_i \cap V_{\mathsf{mul}}) \cup r^{-1}(B_i \cap V_{\mathsf{mul}})) \cap A_i\right\}\right| .$$

Intuitively, the width corresponds to the maximum number of dependencies of the $|V|-i$ largest gates on the i smallest gates for $i \in \{1, \ldots, |V|-1\}$. Note that the width highly depends on the chosen topological order, i.e., two circuits, which are identical up to the linear order, may have different width.

Given a semigroup S, an *input sequence for* $\mathcal{C}$ *over* S is a mapping $\mathsf{in}\colon V_{\mathsf{in}} \to S$. Note that the term *sequence* is justified by the fact that, since V is linearly ordered, there is a canonical bijection between the set of all functions from a fixed subset $W \subseteq V$ to S and the set of sequences $S^{|W|}$. Thus, if the reference to $\mathcal{C}$ is clear from the context and $\mathcal{C}$ has input arity k, we often also use the usual notation $(s_1, \ldots, s_k)$ to describe an input sequence.

Let $\mathcal{C} = (V_{\mathsf{in}}, V_{\mathsf{mul}}, <, \ell, r)$ be a Cayley circuit of size m, let S be a semigroup and let $\mathsf{in}\colon V_{\mathsf{in}} \to S$ be an input sequence. The *sequence computed by* $\mathcal{C}$ *on the input sequence* in is the mapping $\mathsf{out}\colon V \to S$ defined by

$$\mathsf{out}(v) = \begin{cases} \mathsf{in}(v) & \text{if } v \in V_{\mathsf{in}}, \\ \mathsf{out}(\ell(v)) \cdot \mathsf{out}(r(v)) & \text{if } v \in V_{\mathsf{mul}}. \end{cases}$$

If we do not care about the order and multiplicity of the elements in this sequence, we often also refer to $\mathsf{out}(V)$ as the *set of elements computed by* $\mathcal{C}$ *on the input sequence* in. For subsets X and Y of S, we also say that $\mathcal{C}$ *computes* Y *on input* X if Y is contained in the set of elements computed by $\mathcal{C}$ on some input sequence $(s_1, \ldots, s_k) \in X^k$. For convenience, we sometimes identify elements $x \in S$ with the singleton sets $\{x\}$. Whenever a Cayley circuit is only considered for some fixed semigroup S, it is also referred to as *straight-line program over* S (*SLP over* S for short).

It is well-known that powers of semigroup elements can be computed by succinct SLPs. Since the structure of such SLPs only depends on the exponent and not on the semigroup, this idea can also be formulated for Cayley circuits.

Lemma 3.12. *Let $e \in \mathbb{N} \setminus \{0\}$. Then, there exists a Cayley circuit which has a single input gate, has size at most $2\log(e)+1$, has width at most 2, and, given any semigroup S and any element $s \in S$, computes s^e on input s.*

Proof. We prove the statement by induction on e. For $e = 1$, we can use a Cayley circuit with a single input gate. For $e \geqslant 2$, we distinguish two cases. If e is even, induction yields a Cayley circuit of size at most $2\log(\frac{e}{2}) + 1 = 2\log(e) - 1$ computing $s^{e/2}$. We add a single multiplication gate to square this element. If e is odd, we compute $s^{(e-1)/2}$ by a Cayley circuit of size at most $2\log(\frac{e-1}{2}) + 1 \leqslant 2\log(e) - 1$, use a multiplication gate to square the result and then feed the output of this gate into another multiplication gate whose second predecessor is the input gate. □

One can extend the definition of Cayley circuits and also allow *powering gates*, where each powering gate v is assigned a single predecessor $p(v) \in V$ with $p(v) < v$ and a natural number $e(v) \in \mathbb{N}$. The value computed by such a gate is $\mathsf{out}(v) := (\mathsf{out}(p(v)))^{e(v)}$. Once the semigroup and the input sequence are fixed, such extended Cayley circuits can be converted into equivalent regular Cayley circuits with only a factor-$2\log|S|$ blow-up in size and a constant additive blow-up in width.

Proposition 3.13. *Let S be a finite semigroup, let $X \subseteq S$ and $t \in S$. Suppose that there exists an extended SLP with powering gates over S which has size m, width w and computes t on input X. Then, there also exists an (regular) SLP over S which has size at most $2m\log|S|$, width at most $w + 1$ and computes t on input X.*

Proof. Let $\mathcal{C}$ be an SLP over S of input arity k and let $(s_1, \dots, s_k) \in X^k$ be an input sequence for $\mathcal{C}$ such that t is in the set of elements computed by $\mathcal{C}$ on input $(s_1, \dots, s_k)$.

By the pigeon hole principle, every power of the form s^e is equivalent to some power s^f with $f \in \{1, \dots, |S|\}$. Therefore, each exponent $e(v)$ assigned to a powering gate can be replaced by an exponent $f(v) \leqslant |S|$ without changing the sequence computed by $\mathcal{C}$ on input $(s_1, \dots, s_k)$. By Lemma 3.12, every powering gate can be replaced by a width-2 subcircuit of size $2\log|S|$ — note that when using the circuit from Lemma 3.12 as a subcircuit, we do not need its input gate. Since each of these subcircuits does not interact with any other parts of the original circuit, this process does not increase the width of the entire circuit by more than 1. □

The previous proposition works for arbitrarily large exponents. However, as opposed to Lemma 3.12, it requires the semigroup and the inputs to be fixed.

For efficient computations on Cayley circuits, we need the circuits to be given in a suitable encoding. In an *admissible encoding* of a Cayley circuit $\mathcal{C}$ of size m and input arity k, the gates are identified by the natural numbers $\{1, \dots, m\}$ such that for every pair of gates (v, w), the gate v is assigned a smaller number than w if and only if $v < w$. The circuit is then encoded as a list of m fixed-width blocks where

- the first bit of the i-th block specifies whether the i-th gate is an input gate or a multiplication gate and
- in the case of a multiplication gate, the first bit is followed by the numbers of the left and right predecessors.

The width of each block is in $\mathcal{O}(\log m)$. As usual, we assume that each block is padded such that the offset of the first bit of each block is $\mathsf{DLOGTIME}$-computable.

We first investigate the complexity of Cayley circuit evaluation.

Theorem 3.14. *There exists a deterministic random-access Turing machine which, given an admissible encoding of a Cayley circuit $\mathcal{C}$ of input arity k, the Cayley table of a finite semigroup S and elements $s_1, \dots, s_k \in S$, writes the sequence computed by $\mathcal{C}$ on input $(s_1, \dots, s_k)$ on a work tape and runs in time $\mathcal{O}(m^2 \log N)$ where m is the size of $\mathcal{C}$ and N is the cardinality of S. This still works if the Cayley circuit and the input sequence are given on sequential tapes.*

Proof. We preprocess the input by copying both the Cayley circuit $\mathcal{C}$ and the input sequence $(s_1, \dots, s_k)$ to separate work tapes and moving the read-write heads to the first bit on each tape in total time $\mathcal{O}(m \log m + m \log N)$. In the following, we will assume that both the Cayley circuit and the input sequence are provided on separate sequential tapes; only the Cayley table of the semigroup is assumed to be given on the random-access input tape.

The Turing machine scans the types and the incoming edges of each gate of $\mathcal{C}$ in increasing order and computes the corresponding output values one after another. The sequence computed by $\mathcal{C}$ (henceforth called *output sequence*) is written on a third tape and encoded using $\mathcal{O}(m \log N)$ bits. If the currently processed gate is an input gate, the machine obtains the corresponding element from the input sequence in time $\mathcal{O}(\log N)$ and proceeds to the next value on the input sequence tape. If the currently processed gate is a multiplication gate, it obtains the addresses of the two predecessors in time $\mathcal{O}(\log m)$. It then fetches the already computed values of the predecessors in time $\mathcal{O}(m \log N)$ and performs a random-access table lookup to obtain the product of these values in time $\mathcal{O}(\log N)$. In any case, the time needed to process a single gate is in $\mathcal{O}(m \log N)$ which yields a total running time in $\mathcal{O}(m^2 \log N)$. □

A similar time bound holds when asking for the *existence* of an SLP of a given size. The only difference is that the Turing machine needs non-determinism.

Theorem 3.15. *Let $f\colon \mathbb{N} \to \mathbb{N}$ be a time-constructible function. Then there exists a non-deterministic random-access Turing machine which runs in time $\mathcal{O}((f(n))^2 \log n)$ and, given the Cayley table of a finite semigroup S, a set $X \subseteq S$ and an element $t \in S$, accepts if and only if there exists an SLP of size $f(n)$ over S which computes t on input X. Here, n denotes the total input length.*

Proof. The Turing machine first computes the unary encoding of $f(n)$ in time $\mathcal{O}(f(n))$. It then non-deterministically guesses an admissible encoding of a Cayley circuit $\mathcal{C}$ of size at most $f(n)$ in time $\mathcal{O}(f(n) \log f(n))$. The size bound is enforced by advancing a pointer on the unary encoding of $f(n)$ each time a new gate is processed. On a separate work tape, the machine keeps track of the input arity k of the constructed circuit. Afterwards, the machine non-deterministically guesses a sequence $(s_1, \dots, s_k)$ of k elements to copy from X. This can be done in time $\mathcal{O}(f(n) \log n)$. Using the construction from Theorem 3.14, it then verifies that $\mathcal{C}$ computes t on input $(s_1, \dots, s_k)$ in time $\mathcal{O}((f(n))^2 \log n)$. □

In view of Theorem 2.13, an immediate corollary of the previous results is that computability by polylogarithmic-size SLPs is decidable in $\mathsf{PolyLH} \subseteq \mathsf{qAC}^0$. As a complementary result, we now show that computability by polylogarithmic-size SLPs of *bounded width* is also decidable in FOLL. The idea is to use a divide and conquer strategy, repeatedly splitting the circuit into subcircuits of roughly equal size and "guessing" the values at the cut. Since the circuit has bounded width, there are only constantly many values to guess in each splitting step.

Theorem 3.16. *Let $f\colon \mathbb{N} \to \mathbb{N}$ be a function and let $w \in \mathbb{N}$. Then, the problem of deciding, given the Cayley table of a finite semigroup S, a set $X \subseteq S$ and an element $t \in S$, whether there exists an SLP of size $f(n)$ and width at most w over S, which computes t on input X, is decidable by a family of unbounded fan-in Boolean circuits of size $\mathcal{O}(n^{3w} \log f(n))$ and depth $\mathcal{O}(\log f(n))$. Here, n denotes the total input length. If there is a deterministic random-access Turing machine which computes the binary encoding of $f(n)$ on input 1^n, the circuit family is $\mathsf{DLOGTIME}$-uniform.*

Proof. Without loss of generality, we may assume that $f(n) \leqslant n$ for all $n \in \mathbb{N}$. Moreover, for simplicity, we assume that $f(n)$ is a power of 2 for all $n \in \mathbb{N}$. With some additional considerations, the arguments given below also work for arbitrary functions.

We introduce a predicate $P(z_1, \ldots, z_w, y_1, \ldots, y_w, i)$ which is true if there exists an SLP of width at most w and size at most 2^i with w additional input gates and w additional *passthrough gates* (which have in-degree 1 and replicate the value of their predecessors), such that the elements $y_1, \ldots, y_w \in S$ occur as values of the passthrough gates when using $z_1, \ldots, z_w \in S$ as values for the additional input gates and using any subset of the original inputs X as values for the remaining input gates. The additional input gates (resp. passthrough gates) are not counted when measuring the size of the SLP but are considered as input gates (resp. multiplication gates) when measuring width and they have to be the first (resp. last) gates in the vertex ordering. The restriction of the domain of this predicate to a fixed number i contains n^{2w} elements.

The truth value of the predicate for fixed arguments and $i = 0$ can be computed by a constant-depth unbounded fan-in Boolean circuit of size $\mathcal{O}((wn)^2 \log n)$. This is achieved by computing all binary products of the elements $z_1, \ldots, z_w$. Then, each of the values $y_1, \ldots, y_w \notin \{z_1, \ldots, z_w\}$ is verified to be equal to one of these products or to some element of the input set X. For $i \geqslant 1$, the predicate $P(z_1, \ldots, z_w, y_1, \ldots, y_w, i)$ is true if and only if there exist $z'_1, \ldots, z'_w \in S$ such that both $P(z_1, \ldots, z_w, z'_1, \ldots, z'_w, i-1)$ and $P(z'_1, \ldots, z'_w, y_1, \ldots, y_w, i-1)$ are true. Thus, assuming that all truth values of the predicate for $i-1$ are already computed, we can compute $P(z_1, \ldots, z_w, y_1, \ldots, y_w, i)$ with n^w gates in depth 1 because there are only n^w different vectors $(z'_1, \ldots, z'_w) \in S^w$.

Therefore, it suffices to compute the predicates for increasing values of i, until i exceeds the logarithm of $f(n)$ and then, return $P(x, \ldots, x, t, \ldots, t, i)$ for the element t given in the input and for an arbitrary element $x \in X$. This requires $\lceil \log f(n) \rceil$ layers of computation where each layer contains $\mathcal{O}(n^{3w})$ gates and has constant depth.

It is easy to see that the gate numbers can be chosen such that the constructed circuit family is $\mathsf{DLOGTIME}$-uniform (for the first layer, this follows from the observation that a single multiplication can be carried out in $\mathsf{DLOGTIME}$). □

3.3 Computing Powers, Indices and Periods

Combining Theorem 3.14 with the fact that, on input $e \in \mathbb{N}$, one can efficiently construct the Cayley circuit described in Lemma 3.12, one immediately obtains that given a finite semigroup S, elements $s, t \in S$ and an exponent $e \in \{1, \ldots, |S|\}$, the problem of deciding whether $s^e = t$ is in $\mathsf{DPOLYLOGTIME}$. A better bound is obtained by directly using the algorithm for fast exponentiation.

Theorem 3.17. *The problem of deciding, given a finite semigroup S, two elements $s, t \in S$ and an exponent $e \in \{1, \ldots, |S|\}$, whether $s^e = t$ is in $\mathsf{DTIME}(\log^2 n)$. The problem of deciding, given a finite semigroup S, an element $s \in S$ and an integer $i \in \{1, \ldots, |S|\}$, whether s has index i is in $\mathsf{NTIME}(\log^2 n)$. The problem of deciding, given a finite semigroup S, an element $s \in S$ and an integer $p \in \{1, \ldots, |S|\}$, whether s has period p is in $\mathsf{coNTIME}(\log^2 n)$.*

Proof. In order to compute the power s^e, we can use fast exponentiation which requires at most $\log(e)$ square operations and at most $\log(e)$ multiplications of an intermediate result with s. Each square operation and each multiplication is a single table lookup and can be performed in logarithmic time.

To test whether s has index i, we first non-deterministically guess a natural number $p \in \{1, \ldots, |S|\}$ and compute the elements s^{i-1}, s^i and s^p as above. We then verify that $s^i s^p = s^i$ and that $s^{i-1} s^p \neq s^{i-1}$.

To test whether the period of s is not p, we check whether $s^{|S|} s^p = s^{|S|}$. If this test fails, we accept. Otherwise, we guess some element $q \in \{1, \ldots, p-1\}$ and we verify that $s^{|S|} s^q = s^{|S|}$. □

In a fashion similar to Theorem 3.16, we can also implement the computation of powers, indices and periods in polynomial-size circuits of depth $\mathcal{O}(\log \log n)$.

Theorem 3.18. *The problem of deciding, given a finite semigroup S, two elements $s, t \in S$ and an exponent $e \in \{1, \ldots, |S|\}$, whether $s^e = t$ is in FOLL. The problem of deciding, given a finite semigroup S, an element $s \in S$ and an integer $i \in \{1, \ldots, |S|\}$, whether s has index i (resp. period i) is in FOLL.*

Proof. By Proposition 3.13 and using a construction as in Theorem 3.16, we can build a $\mathsf{DLOGTIME}$-uniform family of Boolean circuits of polynomial size and depth $\mathcal{O}(\log \log n)$ computing all powers of all elements of S for all exponents $\{1, \ldots, |S|\}$.

To test whether $s^e = t$, we select the e-th power of s and compare it to t.

To test whether the index of s is i, we select the powers s^{i-1} and s^i and check that there exists a power s^p such that $s^i s^p = s^i$ but $s^{i-1} s^p \neq s^{i-1}$.

To test whether the period of s is i, we verify that $s^{|S|} s^i = s^{|S|}$ and $s^{|S|} s^j \neq s^{|S|}$ for all $j \in \{1, \ldots, i-1\}$. □

The first-order logic introduced in Section 3.1 can be extended by a *power predicate* $x^e = y$. To this end, we need to introduce two types of variables: the first type of variables corresponds to elements of a (partial) finite semigroup S as described in the definition of FO$[\,\cdot\,]$ and the second type of variables corresponds to integers from $\{1, \ldots, |S|\}$. This formalism allows for using formulas such as $\exists e \colon x^e = y \wedge y^2 = y$. One can also add *index* and *period predicates*. By Theorem 3.1, Theorem 3.17 and Theorem 3.18, the truth value of these extended formulas can be computed in both qAC^0 and FOLL.

Chapter 4

The Circuits Properties Framework

This section is devoted to studying compressibility measures for certain representations of elements within classes of finite semigroups. We will introduce two measures associated to each class of finite semigroups $\mathbf{C}$: the first measure concerns the compressibility of products of semigroup elements over arbitrary generators, and the second measure considers a similar property for direct products and measures the compressibility in terms of the sum of the individual components appearing in the direct product. These measures play an essential role in the complexity of many decision problems for recognizable languages when those languages are represented by morphisms to finite semigroups. See Section 5.1.1 and Section 5.4 for applications.

Let $\mathbf{C}$ be a class of finite semigroups and let $f \colon \mathbb{N} \to \mathbb{N}$ be a monotonically increasing function. For convenience, we will only consider time-constructible functions. We say that $\mathbf{C}$ has the $f(n)$ *circuits property* if for each finite semigroup $S \in \mathbf{C}$, for every set $X \subseteq S$ and for each $t \in \langle X \rangle$, there exists an SLP of size at most $f(|S|)$ over S computing t on input X. We say that $\mathbf{C}$ has the $f(n)$ *product circuits property* if for all finite semigroups $S_1, \dots, S_k \in \mathbf{C}$, for every set $X \subseteq S_1 \times \cdots \times S_k$ and for each $t \in \langle X \rangle$, there exists an SLP of size at most $f(|S_1| + \cdots + |S_k|)$ over $S_1 \times \cdots \times S_k$ computing t on input X. For a class of functions $\mathcal{F}$, we say that $\mathbf{C}$ has the $\mathcal{F}$ *circuits property* (resp. $\mathcal{F}$ *products circuits property*) if $\mathbf{C}$ has the $f(n)$ circuits property (resp. products circuits property) for some $f \in \mathcal{F}$. We will also use the terms

- *constant circuits property* (const *CP*, in short) and *constant product circuits property* (const *PCP*, in short) for the class of constant functions, i.e., the class of all functions of the form $f(n) = c$ for some $c \in \mathbb{N}$,
- *polylogarithmic circuits property* (polylog *CP*, in short) and *polylogarithmic product circuits property* (polylog *PCP*, in short) for the class of polylogarithmic functions, i.e., the class of all functions $f(n) \in \mathcal{O}(\log^c n)$ for some $c \in \mathbb{N}$, and
- *polynomial product circuits property* (poly *PCP*, in short) for the class of polynomial functions, i.e., the class of all functions $f(n) \in \mathcal{O}(n^c)$ for some $c \in \mathbb{N}$.

The terminology is inspired by the fact that having the $f(n)$ circuits property is equivalent to requiring every element of a subsemigroup S from the class to be computable by a Cayley circuit of size $f(n)$ over any set of generators of S. The term *polylogarithmic circuits property* was originally introduced in [Fle18c]. The more general notion of $f(n)$ circuits properties was introduced in [Fle18b].

We will also use the term *bounded-width $f(n)$ circuits property* if there exists some constant $w \in \mathbb{N}$ such that for each finite semigroup $S \in \mathbf{C}$ and for each $s \in S$, there exists a Cayley circuit $\mathcal{C}$ of size at most $f(|S|)$ and width w computing s. The abbreviations *bounded-width* const *CP* and *bounded-width* polylog *CP* will be used for the classes of constant and polylogarithmic functions, respectively.

Circuits properties and product circuits properties can alternatively be defined using only SLPs over finite words.

Proposition 4.1. *Let $\mathbf{C}$ be a class of finite semigroups and let $f\colon \mathbb{N} \to \mathbb{N}$ be a monotonically increasing function. Then $\mathbf{C}$ has the $f(n)$ circuits property if and only if for all finite semigroups $S \in \mathbf{C}$, for all morphisms $h\colon A^+ \to S$ and for all words $u \in A^+$, there exists an SLP of size $f(|S|)$ over A^+ which computes a word $v \in A^+$ on input A such that $h(v) = h(u)$.*

For conciseness, we skip the proof, which is straightforward and very similar to the proof of the analogous statement for product circuits properties.

Proposition 4.2. *Let $\mathbf{C}$ be a class of finite semigroups and let $f\colon \mathbb{N} \to \mathbb{N}$ be a monotonically increasing function. Then $\mathbf{C}$ has the $f(n)$ product circuits property if and only if for all sequences of finite semigroups $S_1, \dots, S_k \in \mathbf{C}$, for all morphisms $h_i\colon A^+ \to S_i$ and for all words $u \in A^+$, there exists an SLP of size $f(|S_1| + \cdots + |S_k|)$ over A^+ which computes a word $v \in A^+$ on input A such that $h_i(v) = h_i(u)$ for $1 \leqslant i \leqslant k$.*

Proof. Suppose that $\mathbf{C}$ has the $f(n)$ product circuits property. Let $S_1, \dots, S_k$ be finite semigroups from $\mathbf{C}$, let $h_i\colon A^+ \to S_i$ be morphisms and let $u \in A^+$. We denote by $h\colon A^+ \to S_1 \times \cdots \times S_k$ the product morphism defined by $h(a) = (h_1(a), \dots, h_k(a))$ for all $a \in A$. The $f(n)$ product circuits property yields an SLP of size at most $f(|S_1| + \cdots + |S_k|)$ which computes $h(u)$ on input $h(A)$. By replacing each input $h(a)$ with a, we can easily reuse the underlying Cayley circuit to compute a word $v \in A^+$ with $h(v) = h(u)$.

Conversely, suppose that the right side of the stated equivalence holds. Let $S_1, \dots, S_k$ be finite semigroups from $\mathbf{C}$, let $X \subseteq S_1 \times \cdots \times S_k$ and let $t \in \langle X \rangle$. We denote by X^+ the free semigroup over X and let $h\colon X^+ \to S_1 \times \cdots \times S_k$ be the evaluation morphism, defined as the identity on X. For $i \in \{1, \dots, k\}$, let $h_i\colon X^+ \to S_i$ be the projection onto the i-th component of h. We obtain an SLP of size at most $f(|S_1| + \cdots + |S_k|)$ which computes a word $v \in X^+$ with $h(v) = t$ on input X. This SLP can also be interpreted over $S_1 \times \cdots \times S_k$ and then computes t on input X. □

In the remainder of this chapter, we investigate circuits properties and products circuits properties of certain classes of finite semigroups. For more context and motivation on studying these properties, we recommend the interested reader to advance to the introductions of Section 5.1.1 and Section 5.4.

4.1 Circuits Properties

Our focus is on the polylog CP. The const CP is a less interesting notion on its own — it is equivalent to requiring that all elements be representable by a product of length $\mathcal{O}(1)$.

Proposition 4.3. *Let $\mathbf{C}$ be a class of finite semigroups. Then $\mathbf{C}$ has the* const *CP if and only if there exists a constant $\ell \in \mathbb{N}$ such that for every semigroup $S \in \mathbf{C}$ and for every set $X \subseteq S$, every element $s \in \langle X \rangle$ can be written as a product of length at most ℓ over X.*

Proof. The direction from right to left is trivial: every product of length at most ℓ can be transformed to a canonical SLP which has size at most $2\ell - 1$. For the converse direction, note that explicitly expanding the computation performed by an SLP of size n yields a product of length at most 2^n over X. □

The next proposition is another easy observation which immediately follows from the fact that, by the standard pumping argument, a class of finite semigroups whose cardinalities are bounded by some upper bound k has the $f_k(n)$ circuits property where f_k denotes the constant function $f_k \colon \mathbb{N} \to \mathbb{N}$ with $f_k(n) = k$ for all $n \in \mathbb{N}$.

Proposition 4.4. *Every finite class of finite semigroups has the* const *CP.*

However, finite classes are not the only classes with const CP. The next proposition describes an infinite family of infinite varieties of finite semigroups, each of which has the const CP.

Proposition 4.5. *For each $k \in \mathbb{N}$, the variety $\mathbb{L}\mathbf{I}_k$ has the $4k$ circuits property.*

Proof. Let S be a finite semigroup, let $X \subseteq S$ and let $t \in S$. Let $\ell \in \mathbb{N}$ be minimal such that $t = x_1 \cdots x_\ell$ for some $x_1, \dots, x_\ell \in X$. If ℓ were greater than $2k$, then the equations for $\mathbb{L}\mathbf{I}_k$ yield $t = x_1 \cdots x_k x_{\ell-k+1} \cdots x_\ell$, contradicting the choice of ℓ. A product of length $2k$ can be computed by an SLP of size $4k$. □

In order to prove lower bounds for circuits properties, we shall often use the following simple lemma.

Lemma 4.6. *Let $\mathbf{C}$ be a class of finite semigroups and let $f \colon \mathbb{N} \to \mathbb{N}$ be a monotonically increasing function. Let $S \in \mathbf{C}$, let $X \subseteq S$ with $|X| > f(|S|)$ and let $t \in \langle X \rangle$ such that $t \notin \langle Y \rangle$ for all strict subsets Y of X. Then $\mathbf{C}$ does not have the $f(n)$ circuits property.*

Proof. Suppose, by contradiction, that $\mathbf{C}$ has the $f(n)$ circuits property. Then there exists an SLP of size at most $f(|S|)$ which computes t on input X. Let Y be the set of elements of X assigned to the input gates of this SLP to compute t. Since this SLP has at most $f(|S|) < |X|$ input gates, we have $Y \subsetneq X$. This contradicts the assumption that t does not belong to the subsemigroup of S generated by Y. □

We remark that each class $\mathbb{L}\mathbf{I}_k$ is a subset of $\mathbb{L}\mathbf{I}$, so all varieties considered in Proposition 4.5 do not contain any non-trivial monoids. Using the previous lemma, it is actually easy to show that varieties containing non-trivial monoids cannot have the const CP.

Proposition 4.7. *Let $\mathbf{C}$ be a class of finite semigroups which is closed under taking direct products and contains a non-trivial monoid. Then, $\mathbf{C}$ does not have the $o(\log n)$ circuits property.*

Proof. Let $M \in \mathbf{C}$ be a non-trivial monoid and let $m \neq 1$ be an element of M. For $n \in \mathbb{N}$, we consider the direct product M^n of n copies of M. Let X be the set of elements of M^n which have exactly one of the components set to m and the remaining components set to 1. Clearly, the element $(m, \dots, m)$ is in the subsemigroup of M^n generated by X and does not belong to any semigroup $\langle Y \rangle$ with $Y \subsetneq X$. Moreover, M^n has size $|M|^n$. By Lemma 4.6, $\mathbf{C}$ does not have the $\lfloor \log_{|M|} n \rfloor - 1$ circuits property. □

The statement of the previous proposition does not generalize to arbitrary classes of finite semigroups (which are not necessarily closed under direct products). For example, we already mentioned that every finite class of finite monoids has the const CP.

In the presence of monoids, the polylog CP is of particular interest. We start by investigating the case of finite groups. Circuits properties for finite groups have already been considered by Babai and Szemerédi in a different context [BS84]. For completeness, we restate their result using our terminology and give a fully self-contained proof. Our statement is also slightly stronger than the original lemma since our definition of SLPs does not allow for taking inverses, yet we obtain the same asymptotic upper bound — a naïve corollary of the original lemma only yields SLPs of size $\mathcal{O}(\log^3 n)$.

Lemma 4.8 (Babai-Szemerédi Reachability Lemma). *The variety of finite groups $\mathbf{G}$ has the $\mathcal{O}(\log^2 n)$ circuits property.*

Proof. Let G be a finite group and let $X \subseteq G$. Without loss of generality, we may assume that X is a set of generators for G, otherwise we replace G by $\langle X \rangle$. We construct an SLP of size $\mathcal{O}(\log^2 |G|)$ such that every element of G can be represented as a product of length at most $2\lceil \log |G| \rceil$ over elements computed by the SLP on input X. To this end, we iteratively define a sequence of elements $g_1, \dots, g_\ell \in G$ and a sequence of sets $C_0, \dots, C_\ell \subseteq G$ as follows:

- $C_0 = \{1\}$,
- If $C_i^{-1} C_i = G$, then $\ell = i$ and we are done. Otherwise, we have $C_i^{-1} C_i G \not\subseteq C_i^{-1} C_i$ and thus, $C_i^{-1} C_i X \not\subseteq C_i^{-1} C_i$. Hence, we can choose $y_i \in C_i^{-1}$, $z_i \in C_i$ and $x_i \in X$ such that $y_i z_i x_i \notin C_i^{-1} C_i$, and we let $g_{i+1} = y_i z_i x_i$ and $C_{i+1} = C_i \cup C_i g_{i+1}$.

First note that, for all $i \geqslant 0$, the sets C_i and $C_i g_{i+1}$ are disjoint: if, for some i, the intersection $C_i \cap C_i g_{i+1}$ were non-empty, we would have $g_{i+1} \in C_i^{-1} C_i$, contradicting the choice of g_{i+1}. Thus, each of the unions defining the sets C_i are disjoint unions and we obtain $|C_i| = 2^i$ by induction on i. Consequently, the length ℓ of the sequence is at most $\lceil \log |G| \rceil$. The constructed SLP will be considered on input $\{x_1, \dots, x_\ell\}$. By Proposition 3.13, there is an SLP of size at most $2\ell \log |G|$ which computes the inverses of each of the elements $x_1, \dots, x_\ell$. Note that since G is a group, we have $g^{-1} = g^{|G|-1}$ for all $g \in G$. We may therefore assume in the remainder of the proof that the values $x_1, \ldots, x_\ell$ and their inverses are already precomputed. The gates of this precomputation step will not be counted towards the size bounds obtained below.

By the definition of C_i, we can factorize every element from C_i as $g_1^{\varepsilon_1} \cdots g_i^{\varepsilon_i}$ for some $\varepsilon_1, \ldots, \varepsilon_i \in \{0, 1\}$. Therefore, it suffices to show that $g_1, \ldots, g_\ell$ and their inverses can be computed by an SLP of size $\mathcal{O}(\log^2 |G|)$. The claim then follows with $C_\ell^{-1} C_\ell = G$.

We show that, more generally, for each $i \geqslant 0$, there exists an SLP S_i of size at most $2i(i+1)$ such that each of the elements $g_1, \ldots, g_{i+1}$ and their inverses appear in the sequence computed by S_i. The proof is by induction on i.

For $i = 0$, note that $g_1 = x_0$ and $g_1^{-1} = x_0^{-1}$ are already precomputed. Let now $i \geqslant 1$. We extend S_{i-1} to obtain an SLP which also computes g_{i+1} and g_{i+1}^{-1}. We can factorize $z_i = g_1^{\varepsilon_1} \cdots g_i^{\varepsilon_i}$ with $\varepsilon_1, \ldots, \varepsilon_i \in \{0, 1\}$ and $y_i = g_i^{-\gamma_i} \cdots g_1^{-\gamma_1}$ with $\gamma_1, \ldots, \gamma_i \in \{0, 1\}$. By the induction hypothesis, each of the elements $g_1, \ldots, g_i$ and their inverses appear in the sequence computed by S_{i-1}. Thus, we can add $2i - 1$ multiplication gates to S_{i-1} to obtain an SLP that also computes $y_i z_i$. With another multiplication gate, we then compute $g_{i+1} = y_i z_i x_i$. Similarly, with a total of $2i$ multiplication gates, we can first compute $z_i^{-1} = g_i^{-\varepsilon_i} \cdots g_1^{-\varepsilon_1}$, then $y_i^{-1} = g_1^{\gamma_1} \cdots g_i^{\gamma_i}$ and finally $g_{i+1}^{-1} = x_i^{-1} z_i^{-1} y_i^{-1}$. The size of the resulting SLP is $|S_{i-1}| + 4i \leqslant 2(i-1)i + 4i = 2i(i+1)$, as desired. □

Using a different technique, one can also show that the variety of finite commutative semigroups has the polylog CP.

Lemma 4.9. *The variety of finite commutative semigroups* **Com** *has the bounded-width $\mathcal{O}(\log^2 n)$ circuits property.*

Proof. Suppose that S is a non-trivial commutative semigroup and let X be a set of generators for S. Let $t \in S$ be an arbitrary element. We choose $k \in \mathbb{N}$ to be the smallest value such that there exist elements $x_1, \ldots, x_k \in X$ and integers $i_1, \ldots, i_k \in \mathbb{N}$ with $t = x_1^{i_1} \cdots x_k^{i_k}$. Assume, for the sake of contradiction, that $k \geqslant \log |S| + 1$.

The non-empty elements of the power set $\mathcal{P}(\{1, \ldots, k\})$ form a semigroup T when equipped with set union as binary operation. Consider the morphism $h \colon T \to S$ defined by $h(\{j\}) = x_j^{i_j}$ for all $j \in \{1, \ldots, k\}$. This morphism is well-defined because S is commutative.

Since $|T| = 2^k - 1 \geqslant 2^{\log |S| + 1} - 1 = 2\,|S| - 1 > |S|$, we know by the pigeon hole principle that there exist two sets $K_1, K_2 \subseteq \{1, \ldots, k\}$ with $K_1 \neq K_2$ and $h(K_1) = h(K_2)$. We may assume, without loss of generality, that there exists some $j \in K_1 \setminus K_2$. Now, because

$$t = h(\{1, \ldots, k\}) = h(K_1) h(\{1, \ldots, k\} \setminus K_1) = h(K_2) h(\{1, \ldots, k\} \setminus K_1)$$

and since neither K_2 nor $\{1, \ldots, k\} \setminus K_1$ contain j, we know that t can be written as a product over elements x_i with $1 \leqslant i \leqslant k$ and $i \neq j$, contradicting the choice of k. Proposition 3.13 yields an SLP of size at most $2k(\log |S| + 1) \in \mathcal{O}(\log^2 |S|)$ and width at most 3. □

The previous lemmas can be combined to extend the Babai-Szemerédi Reachability Lemma to Clifford semigroups.

Lemma 4.10. *The variety of finite Clifford semigroups* $\mathbf{G} \vee \mathbf{J_1}$ *has the* $\mathcal{O}(\log^2 n)$ *circuits property.*

Proof. Let $S \in \mathbf{G} \vee \mathbf{J_1}$ be a finite semigroup, let X be a set of generators for S and let $t \in S$. Then t can be written as a product of the form $x_1 \cdots x_\ell$ with $x_1, \dots, x_\ell \in X$.

By Theorem 2.12, the ω-identities $x^{\omega+1} = x$ and $x^\omega y = yx^\omega$ hold in S. Therefore, $t = t^{\omega+1} = t^\omega x_1 \cdots x_k = t^\omega x_1 t^\omega \cdots t^\omega x_k t^\omega$. Theorem 2.12 also states that $(xy)^\omega = x^\omega y^\omega$ holds, so $t^\omega = x_1^\omega \cdots x_k^\omega$.

Each of the elements $x_1^\omega, \dots, x_k^\omega$ can be computed by an SLP of size $\mathcal{O}(\log |S|)$ on input X and since $E(S)$ forms a commutative subsemigroup of S, Lemma 4.9 implies that the element t^ω can be computed by an SLP of size $\mathcal{O}(\log^2 |S|)$ on input $\{x_1^\omega, \dots, x_k^\omega\}$. A better bound for the SLP for t^ω is obtained by looking into the proof of Lemma 4.9: if all generators are idempotent, the resulting SLP has size $\mathcal{O}(\log |S|)$. Together, this yields an SLP of size $\mathcal{O}(\log^2 |S|)$ computing t^ω on input X. Thus, we can also compute $t^\omega X t^\omega$ by an SLP of size $\mathcal{O}(\log^2 |S|)$.

Since $(t^\omega x_i t^\omega)^\omega = t^\omega x_i^\omega = x_1^\omega \cdots x_k^\omega = t^\omega$, the subsemigroup of S generated by $t^\omega X t^\omega$ is a group and by Lemma 4.8, the element t is computed by an SLP of size $\mathcal{O}(\log^2 |S|)$ on input $t^\omega X t^\omega$. □

The results above give rise to the question of whether there is a nice algebraic characterization of the classes of semigroups with the const CP or with the polylog CP. Unfortunately, the upcoming result makes this very unlikely.

Lemma 4.11. *For every integer* $n \geqslant 2$*, there exist a semigroup* S *of size* $\frac{1}{2}(n^2 - n) + 1$ *satisfying* $x^2 = xyx = 0$*, a set* $X \subseteq S$ *with* $|X| = n - 1$ *and an element* $t \in \langle X \rangle$ *such that* $t \notin \langle Y \rangle$ *for all* $Y \subsetneq X$.

Proof. Let $n \geqslant 2$ be a positive integer and let $S = \{(i,j) \mid 1 \leqslant i < j \leqslant n\} \cup \{0\}$ be the semigroup with zero element 0 and with the multiplication on the remaining elements defined by

$$(i,j)(k,\ell) = \begin{cases} (i,\ell) & \text{if } j = k, \\ 0 & \text{otherwise.} \end{cases}$$

Then, for all $(i,j), (k,\ell) \in S$, we have $(i,j)(i,j) = (i,j)(k,\ell)(i,j) = 0$. Therefore, S satisfies the equations $x^2 = xyx = 0$. It is easy to see that the only way to write the element $(1,n)$ of S as a product over $X = \{(1,2), \dots, (n-1,n)\}$ is the sequence $(1,2) \cdots (n-1,n)$. □

In particular, by Lemma 4.6, the variety defined by $x^2 = xyx = 0$ does not have the $o(\sqrt{n})$ circuits property. This observation leads to the following statement.

Proposition 4.12. *Let* $\mathcal{F} \subseteq o(\sqrt{n})$ *be a family of functions containing infinitely many constant functions* $f \colon \mathbb{N} \to \mathbb{N}$*. Then, there is no class of finite semigroups* $\mathbf{C}$ *such that for each variety of finite aperiodic semigroups* $\mathbf{V}$*, we have* $\mathbf{V} \subseteq \mathbf{C}$ *if and only if* $\mathbf{V}$ *has the* $\mathcal{F}$ *circuits property.*

Proof. By Proposition 4.5, every class of finite semigroups $\mathbf{C}$, which contains all varieties of finite aperiodic semigroups with the const CP, also contains the union $\bigcup_k \mathbb{L}\mathbf{I}_k$. It is well-known that $\bigcup_k \mathbb{L}\mathbf{I}_k = \mathbb{L}\mathbf{I}$; this follows almost immediately from Lemma 2.1 and the fact that for two idempotent elements $e, f \in E(S)$ and an element $s \in S$ of a semigroup $S \in \mathbb{L}\mathbf{I}$, we have $esf = esfef = ef$. Since every semigroup satisfying $x^2 = xyx = 0$ trivially satisfies the ω-identity $x^\omega y x^\omega = x^\omega$ for $\mathbb{L}\mathbf{I}$, the class $\mathbf{C}$ then necessarily also contains a variety which does not have the $o(\sqrt{n})$ circuits property by Lemma 4.11 and Lemma 4.6. □

While even within the class of *aperiodic semigroups*, there is no hope for identifying a class of semigroups that captures all varieties with the const CP (resp. polylog CP), surprisingly, the situation is entirely different when considering monoids instead of semigroups. In Proposition 4.7, we already saw that varieties of monoids do not have the $o(\log n)$ circuits property. For the polylog CP, we will prove the following dichotomy result.

Theorem 4.13. *Let* $\mathbf{V}$ *be a variety of finite monoids. Then,* $\mathbf{V}$ *has the* polylog *CP if and only if* $\mathbf{V} \subseteq (\mathbf{G} \vee \mathbf{J_1}) \cup \mathbf{Com}$.

We remark that since $(\mathbf{G} \vee \mathbf{J_1}) \cap \mathbf{A} = \mathbf{J_1} \subseteq \mathbf{Com}$, this theorem immediately yields the following corollary for aperiodic monoids.

Corollary 4.14. *Let* $\mathbf{V}$ *be a variety of finite aperiodic monoids. Then,* $\mathbf{V}$ *has the* polylog *CP if and only if* $\mathbf{V} \subseteq \mathbf{Com}$.

The remainder of this section is devoted to proving Theorem 4.13. The proof is split into three lemmas. The first lemma is similar to Lemma 4.11 but for another variety.

Lemma 4.15. *For every integer* $n \geqslant 2$*, there is a semigroup* $S \in \mathbf{R_1}$ *of size* $\frac{1}{2}(n^2 - n)$*, a set* $X \subseteq S$ *with* $|X| = n-1$ *and an element* $t \in \langle X \rangle$ *such that* $t \notin \langle Y \rangle$ *for all* $Y \subsetneq X$.

Proof. Let $n \geqslant 2$ be a positive integer. Define $S = \{(i,j) \mid 1 \leqslant i < j \leqslant n\}$ with the binary operation

$$(i,j)(k,\ell) = \begin{cases} (i, \max\{j,\ell\}) & \text{if } k \leqslant j, \\ (k,\ell) & \text{otherwise.} \end{cases}$$

A straightforward case-by-case verification shows that this operation is associative. Moreover, $(i,j)(i,j) = (i,j)$ and $(i,j)(k,\ell)(i,j) = (i,j)(k,\ell)$ for all $(i,j),(k,\ell) \in S$. Thus, $S \in \mathbf{R_1}$. We claim that the element $(1,n) = (1,2)\cdots(n-1,n)$ cannot be written as a product over a strict subset of $X = \{(i,i+1) \mid 1 \leqslant i < n\}$. To this end, suppose we are given some product over a subset Y of X which evaluates to $(1,n)$. Note that for all $i,j \in \{1,\dots n-1\}$, we have

$$(i,i+1)(j,j+1) = \begin{cases} (i,i+1) & \text{if } j < i+1, \\ (i,j+1) & \text{if } j = i+1, \\ (j,j+1) & \text{if } j > i+1, \end{cases}$$

so by repeatedly removing factors, the product can be transformed to an equivalent product $(i, i+1)(i+1, i+2) \cdots (j, j+1)$ with $1 \leqslant i < j < n$ and with $(k, k+1) \in Y$ for all $k \in \{i, \ldots, j\}$. Note that the only product of this form which evaluates to $(1, n)$ is for $i = 1$ and $j = n-1$. Then, necessarily, $X = Y$. □

Of course, by left-right symmetry, an analogous statement holds for the variety $\mathbf{L_1}$. In particular, by Lemma 4.6, neither $\mathbf{R_1}$ nor $\mathbf{L_1}$ has the $o(\sqrt{n})$ circuits property. This yields a lower bound for varieties containing U_2 or U_2^{op}. The next lemma focuses on non-commutative varieties containing $C_{2,1}$.

Lemma 4.16. *Let M be a non-commutative monoid. Then N^1 divides the direct product $M \times C_{2,1}^1 \times C_{2,1}^1$.*

Proof. Let $x, y \in M$ such that $xy \neq yx$. Let K be the subsemigroup of $M \times C_{2,1}^1 \times C_{2,1}^1$ generated by $\{(1,1,1), (x,a,1), (y,1,a)\}$ and let $h \colon K \to N^1$ be the morphism defined by $h(1,1,1) = 1$, $h(x,a,1) = a$ and $h(y,1,a) = b$. To see that this morphism is well-defined note that the product of more than two non-neutral elements of K always contains the zero element in the second or third component and is mapped to the zero element under h. The element (xy, a, a) is mapped to ab and the element (yx, a, a) is mapped to 0. This also shows that h is surjective. □

Together with previous results, we obtain the following exclusion characterization.

Lemma 4.17. *Let $\mathbf{V}$ be a variety of finite monoids. Then $\mathbf{V} \subseteq (\mathbf{G} \vee \mathbf{J_1}) \cup \mathbf{Com}$ if and only if neither U_2 nor U_2^{op} nor N^1 belong to $\mathbf{V}$.*

Proof. Each of the monoids U_2, U_2^{op} and N^1 is non-commutative and does not belong to $\mathbf{G} \vee \mathbf{J_1}$. For the converse direction, if $\mathbf{V} \not\subseteq \mathbf{G} \vee \mathbf{J_1}$, we know by Theorem 2.12 that either $U_2 \in \mathbf{V}$ or $U_2^{\mathrm{op}} \in \mathbf{V}$ or $C_{2,1}^1 \in \mathbf{V}$. If $C_{2,1}^1 \in \mathbf{V}$ and, additionally $\mathbf{V} \not\subseteq \mathbf{Com}$, we obtain $N^1 \in \mathbf{V}$ from Lemma 4.16. □

We conclude with the proof of Theorem 4.13.

Proof of Theorem 4.13. Let $\mathbf{V}$ be a variety of finite monoids. If $\mathbf{V} \subseteq \mathbf{Com}$, we know by Lemma 4.9 that $\mathbf{V}$ has the polylog CP. If $\mathbf{V} \subseteq \mathbf{G} \vee \mathbf{J_1}$, we know by Lemma 4.10 that $\mathbf{V}$ has the polylog CP.

Suppose now that $\mathbf{V} \not\subseteq \mathbf{Com}$ and $\mathbf{V} \not\subseteq \mathbf{G} \vee \mathbf{J_1}$. Lemma 4.17 implies that $\mathbf{V}$ contains at least one of the monoids U_2, U_2^{op} or N^1. If $U_2 \in \mathbf{V}$, Proposition 2.7 yields that $\mathbf{V}$ contains all monoids from $\mathbf{R_1}$ and $\mathbf{V}$ does not have the polylog CP by Lemma 4.15. Similarly, if $U_2^{\mathrm{op}} \in \mathbf{V}$, then all monoids from $\mathbf{L_1}$ belong to $\mathbf{V}$ and $\mathbf{V}$ does not have the polylog CP. If $N^1 \in \mathbf{V}$, then Proposition 2.7 shows that $\mathbf{V}$ contains all finite monoids satisfying $x^3 = x^2$ and $x^2 y = xyx = yx^2$. In particular, $\mathbf{V}$ contains all monoids of the form S^1 where S is a semigroup such that $x^2 = xyx = 0$ in S. By Lemma 4.11, $\mathbf{V}$ does not have the polylog CP. □

4.2 Product Circuits Properties

Our first results on product circuits properties arise from a close link to circuits properties. By definition, every class with the $f(n)$ product circuits property also has the $f(n)$ circuits property. In the converse direction, a weaker statement holds.

Proposition 4.18. *Let* $\mathbf{C}$ *be a class of finite semigroups which is closed under taking direct products and has the* $f(n)$ *circuits property. Then,* $\mathbf{C}$ *has the* $f(n^n)$ *product circuits property. If* $\mathbf{C}$ *is the closure of a finite class of finite semigroups under taking direct products, then* $\mathbf{C}$ *has the* $f(2^{\mathcal{O}(n)})$ *product circuits property.*

Proof. Let $S_1, \dots, S_k \in \mathbf{C}$, let $X \subseteq S_1 \times \cdots \times S_k$ and let $t \in S_1 \times \cdots \times S_k$. Since $\mathbf{C}$ is closed under taking direct products, the semigroup $S = S_1 \times \cdots \times S_k$ belongs to $\mathbf{C}$ as well, and since $\mathbf{C}$ has the $f(n)$ circuits property, the element t is computed by an SLP of size $f(|S|)$ on input X. The claim now follows immediately from the sequence of inequalities

$$|S| = |S_1| \cdots |S_k| \leqslant \max\{|S_1|, \dots, |S_k|\}^k \leqslant (|S_1| + \cdots + |S_k|)^{|S_1| + \cdots + |S_k|}$$

where the last inequality uses the fact that $|S_i| \geqslant 1$ for each $i \in \{1, \dots, k\}$.

Suppose now that $\mathbf{C}$ is the closure of some finite class $\mathbf{C}'$ under taking direct products. We may assume without loss of generality that each of the semigroups $S_1, \dots, S_k$ is non-trivial and belongs to $\mathbf{C}'$. Thus, there exists a constant C such that $|S_1|, \dots, |S_k| \leqslant C$. We obtain $|S| \leqslant \max\{|S_1|, \dots, |S_k|\}^k \leqslant C^{|S_1| + \cdots + |S_k|} = 2^{\log(C) \cdot (|S_1| + \cdots + |S_k|)}$. □

Applying a polylogarithmic function to the function $n \mapsto n^n$ yields a polynomial in n. Therefore, we obtain the following corollary.

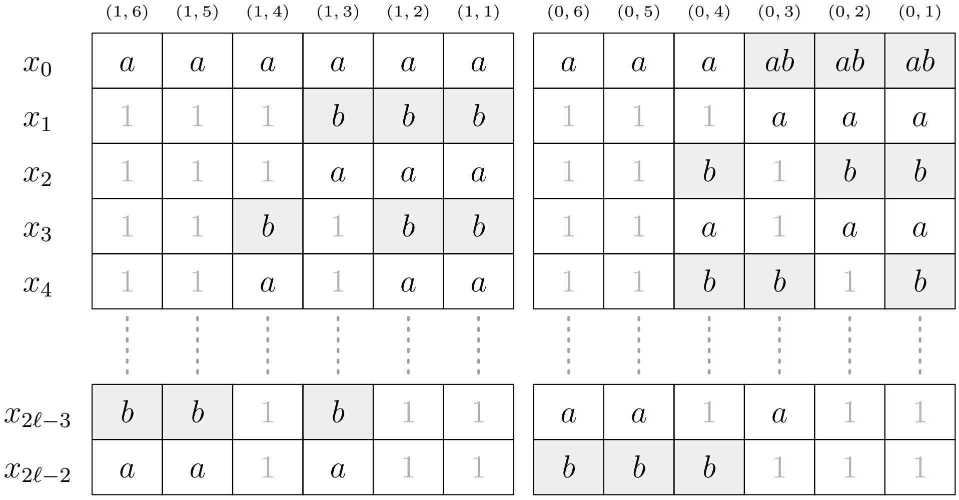
Corollary 4.19. *Let* $\mathbf{C}$ *be a class of finite semigroups which is closed under taking direct products. Then,* $\mathbf{C}$ *has the* const *CP if and only if it has the* const *PCP. Moreover, if* $\mathbf{C}$ *has the* polylog *CP, then* $\mathbf{C}$ *has the* poly *PCP.*

In particular, the varieties $\mathbf{G} \vee \mathbf{J_1}$ and $\mathbf{Com}$ have the poly PCP. Each of the varieties $\mathbb{L}\mathbf{I}_k$ has the const PCP. With the same construction as in Proposition 4.7, one can show that sublinear product circuits properties do not appear in the presence of non-trivial monoids.

Proposition 4.20. *Let* $\mathbf{C}$ *be a class of finite semigroups which contains a non-trivial monoid. Then,* $\mathbf{C}$ *does not have the* $o(n)$ *product circuits property.*

Note that the proof of Proposition 4.7 actually is a proof for the statement of Proposition 4.20, and Proposition 4.7 can be seen as an immediate corollary of Proposition 4.20 and Proposition 4.18.

We stress that the statement of Corollary 4.19 does not generalize to arbitrary classes of finite semigroups (which are not necessarily closed under direct products). For a counterexample, we consider the class $\{B_2^1\}$. It follows from [FK18c] that, assuming $\mathsf{NP} \neq \mathsf{PSPACE}$, this class does not have the poly PCP. In the following, we will give an alternative proof which does not rely on any unproven assumptions. The essential part is captured in the following lemma.



	$(1,6)$	$(1,5)$	$(1,4)$	$(1,3)$	$(1,2)$	$(1,1)$	$(0,6)$	$(0,5)$	$(0,4)$	$(0,3)$	$(0,2)$	$(0,1)$
x_0	a	a	a	a	a	a	a	a	a	ab	ab	ab
x_1	1	1	1	b	b	b	1	1	1	a	a	a
x_2	1	1	1	a	a	a	1	1	b	1	b	b
x_3	1	1	b	1	b	b	1	1	a	1	a	a
x_4	1	1	a	1	a	a	1	1	b	b	1	b
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$x_{2\ell-3}$	b	b	1	b	1	1	a	a	1	a	1	1
$x_{2\ell-2}$	a	a	1	a	1	1	b	b	b	1	1	1

Figure 4.1: The unique sequence of elements evaluating to t (for $n = 6$)

Lemma 4.21. *Let $n \in \mathbb{N}$ be an even integer and let $S = (B_2^1)^{2n}$ be the $2n$-fold direct product of copies of B_2^1. Then, there exist a set $X \subseteq S$ of cardinality $2\binom{n}{n/2} - 1$ and an element $t \in S$ such that $t \in \langle X \rangle$ but $t \notin \langle Y \rangle$ for all $Y \subsetneq X$.*

Proof. Throughout the following construction, let $\ell = \binom{n}{n/2}$ and let $\{R_0, \dots, R_{\ell-1}\}$ be the set of $n/2$-element subsets of $\{1, \dots, n\}$. Note that we do not care about the order of the subsets $R_0, \dots, R_{\ell-1}$ but require them to be pairwise disjoint.

The $2n$-fold direct product S can be interpreted as the set of all functions from $\{1, \dots, n\} \times \{0, 1\}$ to B_2^1 with componentwise multiplication. Using this notation, we define an element $t \in S$ and a set $X = \{x_0, \dots, x_{2\ell-2}\} \subseteq S$ by

$$x_0(i,j) = \begin{cases} ab & \text{if } i \in R_0 \wedge j = 0, \\ a & \text{otherwise,} \end{cases} \qquad t(i,j) = \begin{cases} ab & \text{if } i \in R_{\ell-1} \wedge j = 0, \\ a & \text{otherwise,} \end{cases}$$

$$x_{2k-1}(i,j) = \begin{cases} a & \text{if } i \in R_{k-1} \wedge j = 0, \\ b & \text{if } i \in R_{k-1} \wedge j = 1, \\ 1 & \text{otherwise,} \end{cases} \qquad x_{2k}(i,j) = \begin{cases} b & \text{if } i \in R_k \wedge j = 0, \\ a & \text{if } i \in R_{k-1} \wedge j = 1, \\ 1 & \text{otherwise,} \end{cases}$$

for all $i \in \{1, \dots, n\}$, $j \in \{0, 1\}$ and $k \in \{1, \dots, \ell - 1\}$.

We claim that $x_0 \cdots x_{2\ell-2}$ is the only sequence of elements of X whose product is t in S. This sequence is depicted in Figure 4.1. To prove this claim, we show that for every $k \in \{0, \dots, \ell - 1\}$, there exists exactly one sequence in X^{2k+1} whose product evaluates to some element $s_k \in S$ with $s_k \geqslant_{\mathcal{R}} t$. Moreover, we prove that this sequence is $x_0 \cdots x_{2k}$ and that its evaluation s_k in S corresponds to the function

$$s_k(i,j) = \begin{cases} ab & \text{if } i \in R_k \wedge j = 0, \\ a & \text{otherwise.} \end{cases}$$

The proof is by induction on k. Note that $s_{\ell-1} = t$, as desired. Also note that for each $x \in X$, at least one of the components of the product tx is the zero element of B_2^1.

For $k = 0$, note that since $bB_2^1 = \{b, ba, 0\}$ in B_2^1, any product over X not starting with x_0 contains b or ba or 0 in at least one of the components. In t, however, all components are either a or ab. Thus, any product over X evaluating to t in S must start with x_0. Moreover, $s_0 = x_0$, as desired.

Suppose now that $k \geqslant 1$ and let $(y_0, \ldots, y_{2k})$ be any sequence of elements of X whose product $y_0 \cdots y_{2k}$ is some element $s_k \geqslant_{\mathcal{R}} t$ in S. Then, $y_0 \cdots y_{2k-2} \geqslant_{\mathcal{R}} s_k \geqslant_{\mathcal{R}} t$ and by the induction hypothesis, we obtain $y_m = x_m$ for $m \in \{0, \ldots, 2k-2\}$. Moreover, the product $y_0 \cdots y_{2k-2} = x_0 \cdots x_{2k-2}$ equals s_{k-1} in S. Note that $s_{k-1}(i, 1) = a$ for $1 \leqslant i \leqslant n$. Therefore, if y_{2k-1} were x_m for some even number m, at least one of the components in the product $s_k y_{2k-1}$ would be 0, a contradiction to $s_{k-1} y_{2k-1} y_{2k} \geqslant_{\mathcal{R}} t$. On the other hand, if $y_{2k-1} = x_{2m-1}$ with $m \in \{1, \ldots, \ell - 1\} \setminus \{k\}$, we choose an integer $i \in R_{m-1} \setminus R_{k-1}$. Then, $s_{k-1}(i, 0) = x_{2m-1}(i, 0) = a$, so again, at least one of the components of $s_{k-1} y_{2k-1}$ is 0. Consequently, $y_{2k-1} = x_{2k-1}$. By a similar argument, we obtain $y_{2k} = x_{2k}$.

The product $s_{k-1} x_{2k-1} x_{2k}$ is

$$s_{k-1} x_{2k-1} x_{2k}(i, j) = \begin{cases} a \;\; 1 \;\; 1 & \text{if } i \notin R_{k-1} \wedge i \notin R_k \wedge j = 0, \\ a \;\; 1 \;\; b & \text{if } i \notin R_{k-1} \wedge i \in R_k \wedge j = 0, \\ ab \;\; a \;\; 1 & \text{if } i \in R_{k-1} \wedge i \notin R_k \wedge j = 0, \\ ab \;\; a \;\; b & \text{if } i \in R_{k-1} \wedge i \in R_k \wedge j = 0, \\ a \;\; 1 \;\; 1 & \text{if } i \notin R_{k-1} \qquad\quad \wedge j = 1, \\ a \;\; b \;\; a & \text{if } i \in R_{k-1} \qquad\quad \wedge j = 1, \end{cases}$$

which is easily seen to be the same as s_k, thereby concluding the proof of the claim. □

The following proposition is an immediate consequence of the previous lemma and a variation of Lemma 4.6.

Proposition 4.22. *The class $\{B_2^1\}$ does not have the $\left\lfloor \frac{2^{n+1}}{n} \right\rfloor$ product circuits property.*

Proof. Note that $\frac{2^{n+1}}{n} < 2\binom{n}{n/2} - 1$ for $n \geqslant 4$. The statement now is an immediate consequence of the previous lemma and the fact that an SLP of size m can have at most m input gates. □

By Theorem 2.11, we obtain the following result for varieties of finite semigroups.

Theorem 4.23. *If $\mathbf{V}$ is a variety of finite semigroups with $\mathbf{V} \not\subseteq \mathbb{L}\mathbb{D}\mathbf{S}$, then $\mathbf{V}$ does not have the* poly *PCP. If $\mathbf{V}$ is a variety of finite monoids with $\mathbf{V} \not\subseteq \mathbb{D}\mathbf{S}$, then $\mathbf{V}$ does not have the* poly *PCP.*

According to current knowledge, the monoid B_2^1 is the archetypal example of a semigroup without the poly PCP. It is open whether or not $\mathbb{L}\mathbb{D}\mathbf{S}$ has the poly PCP. However, we do know that the converse of the second part of Corollary 4.19 does not hold, i.e., having the poly PCP does not imply having the polylog CP. For example, the varieties $\mathbf{R}$ and $\mathbf{L}$ have the poly PCP: in a minimal-length representation of an element from the direct product of $\mathcal{R}$-trivial semigroups, every generator causes an $<_{\mathcal{R}}$-descent in at least one of the components. Thus, the length of this product is bounded by the sum of the cardinalities of the individual semigroups forming the direct product. By a very similar argument as in the proof of Proposition 4.5, one can also show that each of the varieties $\mathbb{L}\mathbf{I}_k$ has the $4k$ product circuits property, and deduce that $\mathbb{L}\mathbf{I}$ has the poly PCP [Fle18b]. For further results on the poly PCP, we refer to [FK18c].

From Theorem 4.13, Corollary 4.19 and Proposition 4.20, we know that every non-trivial variety of finite monoids contained in $(\mathbf{G} \vee \mathbf{J_1}) \cup \mathbf{Com}$ has the poly PCP but not the polylog PCP. The previous theorem yields varieties which do not have the poly PCP. We conclude by presenting a family of varieties which have the polylog PCP but do not have the const PCP. Note that for product circuits properties, a statement similar to Proposition 4.12 holds, i.e., there is no maximal class of finite semigroups with the const PCP or the polylog PCP; we refer to Corollary 5.40 for details. Nevertheless, it is useful to identify sufficient conditions for the polylog PCP, since this property will be shown to yield efficient (quasi-polynomial-time) algorithms for intersection non-emptiness. In contrast, we will also show that the intersection non-emptiness problem is NP-hard for all classes of finite semigroups known to not have the polylog PCP, and PSPACE-complete for all classes known to not have the poly PCP.

We say that a class of finite semigroups $\mathbf{C}$ has *unbounded index* if the supremum of all indices of all elements of all semigroups in $\mathbf{C}$ is ∞. Otherwise, the class $\mathbf{C}$ has *bounded index*. In the literature, these classes are sometimes also referred to as classes with *unbounded torsion* and *bounded torsion*, respectively. The proof of the next theorem is similar to the proof of Lemma 4.9 but slightly more involved.

Theorem 4.24. *Let $\mathbf{C} \subseteq \mathbf{Com} \cap \mathbb{L}\mathbf{I}$ be a class of finite semigroups of bounded index. Then, $\mathbf{C}$ has the* polylog *PCP.*

Proof. Let S be a non-trivial semigroup from $\mathbf{Com} \cap \mathbb{L}\mathbf{I}$. We show that if every cyclic subsemigroup of S has size at most k, then every product of length at least $k(\log|S|+1)$ over S evaluates to the zero element. Thus, every product over a set $X \subseteq S$ of length at least $k(\log|S|+1)$ can be truncated after the first $\lceil k(\log|S|+1)\rceil$ elements without changing its value. Note that the ω-identities $x^\omega y = yx^\omega = x^\omega$ hold in $\mathbf{Com} \cap \mathbb{L}\mathbf{I}$, so every element has period 1 and the k-fold power of any element in S is the zero element.

Suppose by way of contradiction that there exists a product of length at least $k(\log|S|+1)$ which is not the zero element. Among all such products, we choose a product where the number of different elements appearing in the product is minimal. We denote this number by m. By rearranging the elements, we can rewrite this product as $s_1^{i_1} \cdots s_m^{i_m}$ with $s_i \neq s_j$ for $1 \leqslant i < j \leqslant m$. If $m \leqslant \log|S|+1$, then there exists some $\ell \in \{1, \dots, m\}$ with $i_\ell \geqslant k$. The element $s_\ell^{i_\ell}$ then is the zero element, a contradiction. Suppose now that $m \geqslant \log|S|+1$.

The set $T = \mathcal{P}(\{1, \dots, m\}) \setminus \{\emptyset\}$ forms a semigroup with union as binary operation. Let $h\colon T \to S$ be the morphism defined by $h(\{\ell\}) = s_\ell^{i_\ell}$ for $1 \leqslant \ell \leqslant m$. We have

$|T| = 2^m - 1 \geqslant 2^{\log|S|+1} - 1 = 2\,|S| - 1 > |S|$. Thus, by the pigeon hole principle, there exist two sets $K_1, K_2 \subseteq \{1, \ldots, m\}$ with $K_1 \neq K_2$ and $h(K_1) = h(K_2)$.

If $K_1 \subsetneq K_2$, then multiplying the product by $h(K_2 \setminus K_1)$ does not change its value and k-fold multiplication shows that the product is zero, a contradiction. The case $K_2 \subsetneq K_1$ is symmetric. Thus, we may assume that neither $K_1 \subseteq K_2$ nor $K_2 \subseteq K_1$. The *length* of a set $K \subseteq \{1, \ldots, m\}$ is the sum of all i_ℓ with $\ell \in K$. By symmetry, we may assume that the length of K_1 is at most the length of K_2. We replace the factor $h(K_1)$ of the product by $h(K_2)$. By $h(K_1) = h(K_2)$, the new product is, again, not the zero element. Since the length of K_1 is at most the length of K_2, the new product still has length at least $k(\log|S| + 1)$. However, since $K_1 \setminus K_2 \neq \emptyset$, less than m pairwise different elements appear in this new product, contradicting the choice of m. □

An example of a variety for finite semigroups with the polylog PCP is the variety defined by the equations $xy = yx$ and $x^2 = 0$. This variety contains only commutative semigroups, is contained in $\mathbb{L}\mathbf{I}$ and has bounded index. It is also easy to see that this variety does not have the const PCP. Consider the semigroup $S = \mathcal{P}(\{1, \ldots, n\}) \setminus \{\emptyset\}$ with the multiplication defined by

$$X \cdot Y = \begin{cases} X \cup Y & \text{if } X \cap Y = \emptyset, \\ \{1, \ldots, n\} & \text{otherwise.} \end{cases}$$

This semigroup is commutative and every square is the zero element $\{1, \ldots, n\}$. Moreover, the element $\{1, \ldots, n-1\}$ of S cannot be written as a product over a strict subset of the set of singletons $\{\{i\} \mid 1 \leqslant i < n\}$.

Chapter 5

Decision Problems

The previous chapters laid the groundwork for efficient algorithms on finite semigroups. In this chapter, we will study common decision problems on finite semigroups and recognizable languages.

5.1 Subsemigroup Membership

The *subsemigroup membership* problems asks, given a semigroup S, a set of generators $X \subseteq S$ and an element $t \in S$, whether t belongs to the subsemigroup of S generated by X. The decidability and complexity of the problem highly depends on the encoding of the input. In this section, we consider the following two variants of the problem.

1. The *membership problem for transformation semigroups*, where both the set of generators X and the element t are given as pointwise transformations on a common finite set Q. The semigroup S is implicit; it is the subsemigroup of the full transformation semigroup on Q generated by X.

2. The *Cayley subsemigroup membership problem*, where the semigroup is given in Cayley encoding as described in Section 2.4.5.

We also consider restrictions of these problems to certain classes of finite semigroups. More formally, let $\mathbf{C}$ be a class of finite semigroups. Then, the *membership problem for transformation semigroups for* $\mathbf{C}$, denoted by Trans–SM($\mathbf{C}$), is defined as follows:

Trans–SM($\mathbf{C}$)	
Input:	Transformations $x_1, \ldots, x_k, t \colon Q \to Q$ with $S = \langle x_1, \ldots, x_k \rangle \in \mathbf{C}$
Question:	Does t belong to S?

The *Cayley subsemigroup membership problem for* $\mathbf{C}$, denoted by Cayley–SM($\mathbf{C}$), is defined as follows:

Cayley–SM($\mathbf{C}$)	
Input:	The Cayley table of a semigroup $S \in \mathbf{C}$, a set $X \subseteq S$ and an element $t \in S$
Question:	Is t in the subsemigroup of S generated by X?

Semigroups	Monoids	Trans–SM	Cayley–SM
Ab	**Ab**	$\mathsf{ZPL}^{\mathsf{ModL}}$, ModL-hard	L, FOLL, NPOLYLOGTIME
G	**G**	NC, NL-hard, ModL-hard	L, NPOLYLOGTIME
NB	$\mathbf{J_1}$	AC^0	$\Pi_2\mathsf{TIME}(\log n)$
RB	**RB**	P-complete	$\Pi_2\mathsf{TIME}(\log n)$
B	**B**	NP-complete	NL
Com	**Com**	NP-complete	FOLL, NPOLYLOGTIME
J	**J**	NP-complete	NL-complete
R	**R**	NP-complete	NL-complete
L	**L**	PSPACE, NP-hard	NL-complete
$\mathbb{LD}\mathbf{S}$	$\mathbb{D}\mathbf{S}$	PSPACE, NP-hard	NL-complete
S	**S**	PSPACE-complete	NL-complete

Table 5.1: The complexity of Trans–SM and Cayley–SM for certain varieties

We use Trans–SM as a shorthand for Trans–SM(**S**) and Cayley–SM as a shorthand for Cayley–SM(**S**). Our main motivation for studying these problems is their close connection to numerous language decision problems. These connections will be established in Sections 5.3 and 5.4.

It is well-known that Trans–SM is PSPACE-complete and that Cayley–SM is NL-complete. The PSPACE-completeness result has been published as part of Kozen's seminal paper on *lower bounds for natural proof systems* [Koz77]. A proof that Cayley–SM is NL-complete was published by Jones, Lien and Laaser in 1976 [JLL76], following earlier work by Jones and Laaser who showed that dropping the associativity requirement of the Cayley table in Cayley–SM yields a P-complete problem [JL76]. Both the PSPACE-completeness result and the NL-completeness result have since been the building block for numerous hardness results in formal language theory and related fields; see e.g. [BM91, GKM06, Yam13, Ber97, CH91, DGH05, JR91]. We anticipate that both results are corollaries of more general theorems presented and proved in the upcoming sections.

Further research focused on investigating the complexity of these problems when additional restrictions are imposed on the inputs. A series of research papers [BLS87, Bea88a, Bea88b, Bea94, BMT92, FHL80, Sim68] extensively studied the complexity of Trans–SM(**C**) for various classes **C** of finite semigroups. The Cayley semigroup membership problem for groups Cayley–SM(**G**) was studied by Barrington, McKenzie, Kadau and Lange starting in the early 1990's [BM91, BKLM01]. Before diving into the investigation of certain classes, let us give an overview of state-of-the-art complexity results on Trans–SM(**C**) and Cayley–SM(**C**).

Table 5.1 illustrates the complexity of both problems for three different types of varieties of finite semigroups: varieties of finite groups, varieties of finite bands and varieties of finite semigroups containing non-band semigroups. The results for transformation monoid membership for aperiodic monoids can be found in [BMT92]. The results on the complexity of Trans–SM for idempotent monoids are particularly pleasing, because there are no intermediate monoid varieties between $\mathbf{J_1}$ and $\mathbf{RB}$, and the complexity results can be subsumed in a trichotomy theorem. This is explicated and extended to semigroups in Section 5.1.3. An NC algorithm for the membership problem in transformation groups was described in [BLS87]. Containment in $\mathsf{ZPL}^{\mathsf{ModL}}$ and ModL-hardness of the membership problem for Abelian permutation groups follows from [AV04]; NL-hardness was established in [MC87].

The complexity results of the right column of Table 5.1 will be proved in the upcoming sections. It is easy to see that, for a given class of finite semigroups, the Cayley semigroup membership problem cannot be harder than the transformation semigroup membership problem. This is a direct consequence of Lemma 2.14.

Proposition 5.1. *Let* $\mathbf{C}$ *be a class of finite semigroups. Then* Cayley–SM($\mathbf{C}$) $\leqslant_{\mathsf{AC}^0}$ Trans–SM($\mathbf{C}$)*.*

Unfortunately, this reduction is not very useful: it is known that for any non-trivial variety of finite aperiodic monoids $\mathbf{V}$ except $\mathbf{J_1}$, the decision problem Trans–SM($\mathbf{V}$) is P-hard, whereas Cayley–SM($\mathbf{V}$) belongs to NL. For groups, Trans–SM($\mathbf{G}$) is NL-hard, whereas Cayley–SM($\mathbf{G}$) belongs to L (and is not hard for L, for NL, or for any other complexity class containing Parity as shown later in this section). The only transfer result obtained from Proposition 5.1 is for the variety $\mathbf{NB}$. However, membership of Cayley–SM($\mathbf{NB}$) to AC^0 also follows from the observation that Cayley–SM($\mathbf{RB}$) belongs to a low level of the logarithmic-time hierarchy.

5.1.1 Clifford Semigroups and Commutative Semigroups

Early research on the Cayley semigroup membership problem focused on the generic case and on the group case. One of the first published insights on the group case is the observation by Barrington and McKenzie that Cayley–SM($\mathbf{G}$) reduces to reachability in *undirected* graphs [BM91]. As a consequence, Cayley–SM($\mathbf{G}$) belongs to the complexity class SL which is nowadays known to be identical to L by Reingold's seminal paper on deciding undirected s-t-connectivity in deterministic log-space [Rei08]. Barrington and McKenzie conjectured that Cayley–SM($\mathbf{G}$) is L-complete, which seemed like a quite natural conjecture given the plethora of L-completeness results on decision problems for finite groups. This conjecture remained unproven for a long time and withstood any counterarguments until in 2001, Barrington, Kadau, Lange and McKenzie showed that Cayley–SM($\mathbf{Ab}$) belongs to the complexity class FOLL, thereby also proving that Cayley–SM($\mathbf{Ab}$) cannot be L-complete. The case of general groups remained open until earlier this year, the author of the present work proved that a similar result also holds for both Cayley–SM($\mathbf{G}$) and Cayley–SM($\mathbf{Com}$): neither of these classes can be hard for any class containing Parity. This proof was presented in [Fle18c]. It leverages the fact that both $\mathbf{G}$ and $\mathbf{Com}$ have the polylog CP and shows that Cayley semigroup membership for classes with the polylog CP is in qAC^0. In this work, the result is generalized in two ways. Firstly, as shown in Chapter 4, one can extend the group case to the variety of finite Clifford semigroups $\mathbf{G} \vee \mathbf{J_1}$. Secondly, the complexity upper bound is improved from qAC^0 to $\mathsf{NPOLYLOGTIME}$. This $\mathsf{NPOLYLOGTIME}$ upper bound essentially already follows from Theorem 3.15 but for completeness, we state it as a corollary here.

Corollary 5.2. *Let* $\mathbf{C}$ *be a class of finite semigroups. If* $\mathbf{C}$ *has the* $f(n)$ *circuits property, then* Cayley–SM($\mathbf{C}$) *is decidable by a non-deterministic random-access Turing machine in time* $\mathcal{O}((f(n))^2 \log n)$. *In particular, the following complexity results hold:*

1. *If* $\mathbf{C}$ *has the* const *CP, then* Cayley–SM($\mathbf{C}$) $\in \mathsf{NLOGTIME} \subseteq \mathsf{AC}^0$.

2. *If* $\mathbf{C}$ *has the* polylog *CP, then* Cayley–SM($\mathbf{C}$) $\in \mathsf{NPOLYLOGTIME} \subseteq \mathsf{qAC}^0$.

Similarly, we obtain the following corollary of Theorem 3.16.

Corollary 5.3. *Let* $\mathbf{C}$ *be a class of finite semigroups. If* $\mathbf{C}$ *has the bounded-width* $f(n)$ *circuits property, then* Cayley–SM($\mathbf{C}$) *is decidable by a family of unbounded fan-in Boolean circuits of size* $\mathcal{O}(n^{3w} \log f(n))$ *and depth* $\mathcal{O}(\log f(n))$ *for some* $w \in \mathbb{N}$. *This family is* $\mathsf{DLOGTIME}$*-uniform if* f *satisfies the condition stated in Theorem 3.16. In particular, if* $\mathbf{C}$ *has the bounded-width* polylog *CP, then* Cayley–SM($\mathbf{C}$) $\in \mathsf{FOLL}$.

With the results from Chapter 4, particularly Lemma 4.9 and Lemma 4.10, we obtain the following complexity bounds:

Corollary 5.4. *The decision problems* Cayley–SM($\mathbf{G} \vee \mathbf{J_1}$) *and* Cayley–SM($\mathbf{Com}$) *belong to* $\mathsf{NPOLYLOGTIME}$. *The decision problem* Cayley–SM($\mathbf{Com}$) *belongs to* FOLL. *In particular, neither* Cayley–SM($\mathbf{G} \vee \mathbf{J_1}$) *nor* Cayley–SM($\mathbf{Com}$) *are hard for any class containing* Parity *(under non-uniform* qAC^0 *reductions).*

It is worth noting that by the standard circuit simulation argument, the class FOLL is contained in $\mathsf{DSPACE}(\log(n) \log\log(n))$. However, we do not know whether or not Cayley–SM($\mathbf{Com}$) can be decided by a Turing machine in deterministic log-space. On the other hand, while Reingold's result on s-t-connectivity yields that Cayley–SM($\mathbf{G}$) is in L, it is open whether Cayley–SM($\mathbf{G}$) belongs to FOLL.

It is also worth mentioning that for varieties of finite monoids, the previous corollary can be extended to obtain a dichotomy result.

Corollary 5.5. *Let* $\mathbf{V}$ *be a variety of finite monoids. Then, the following properties are equivalent:*

1. Cayley–SM($\mathbf{V}$) *belongs to* $\mathsf{NPOLYLOGTIME}$.

2. $\mathbf{V}$ *has the* polylog *CP.*

3. $\mathbf{V} \subseteq (\mathbf{G} \vee \mathbf{J_1}) \cup \mathbf{Com}$.

Proof. The equivalence of (2) and (3) is established in Theorem 4.13. Corollary 5.2 states that (2) implies (1).

To see that (1) implies (3), suppose that $\mathbf{V} \not\subseteq (\mathbf{G} \vee \mathbf{J_1}) \cup \mathbf{Com}$. By Lemma 4.17, we know that $U_2 \in \mathbf{V}$ or $U_2^{\mathrm{op}} \in \mathbf{V}$ or $N^1 \in \mathbf{V}$. In any of these cases, Proposition 2.7, Lemma 4.11 and Lemma 4.15 yield that for every $n \geqslant 2$, there exist a monoid $S \in \mathbf{V}$ of size $\Theta(n^2)$, a set $X \subseteq S$ of size $n-1$ and an element $t \in \langle X \rangle$ such that t cannot be written as a product over a strict subset of X. Suppose there exists a non-deterministic random-access machine accepting the inputs S, X and t in polylogarithmic time. Then, there exists some constant $c \in \mathbb{N}$ such that the machine accesses only $\mathcal{O}(\log^c n)$ bits of X on every accepting path. Thus, for some large enough n, we can remove an element from the set of generators X (or replace it with any other element) without affecting acceptance. This contradicts the choice of S, X and t. □

Note that if $\mathbf{V}$ is a variety of finite monoids with $\mathbf{V} \not\subseteq \mathbf{CR} \cup \mathbf{Com}$, the problem is actually NL-hard already, as we shall see in the next section.

5.1.2 Hardness Results

Before presenting efficient algorithms for other classes, we generalize the lower bounds given in [JLL76]. Using a slightly more sophisticated reduction, which is reminiscent of the proof that reachability in directed *acyclic* graphs is NL-complete, we obtain NL-completeness for a fairly large class of finite semigroups.

Theorem 5.6. *Let* $\mathbf{V}$ *be the variety of finite semigroups defined by the equations* $x^2 = xyx = 0$. *Then, the decision problem* Cayley–SM($\mathbf{V}$) *is* NL*-complete under* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions.*

Proof. We reduce reachability in directed graphs to Cayley–SM($\mathbf{V}$). Let $G = (V, E)$ be a directed graph with n vertices. We define a semigroup

$$S = V \times \{(i,j) \mid 1 \leqslant i < j \leqslant n\} \times V \cup \{0\}$$

with the binary operation

$$(v,i,j,w)(x,k,\ell,y) = \begin{cases} (v,i,\ell,y) & \text{if } w = x \text{ and } j = k, \\ 0 & \text{otherwise.} \end{cases}$$

Let $E' = E \cup \{(v,v) \mid v \in V\}$. It is clear that the element $(s,1,n,t)$ is in the subsemigroup of S generated by

$$X = \{(v,i,j,w) \mid (v,w) \in E' \text{ and } 1 \leqslant i \leqslant n-1 \text{ and } j = i+1\}$$

if and only if t is reachable from s in G. By definition of the multiplication in S, the second components of any sequence of elements of S whose product is a non-zero element has to be strictly monotonically increasing. Therefore, no such sequence contains the same element twice, and the equations $x^2 = xyx = 0$ hold in S.

It is clear that every entry of the Cayley table and every element of X can be computed in $\mathsf{DLOGTIME}$, assuming that the graph is given in a suitable encoding, such as an adjacency matrix with entries whose addresses are $\mathsf{DLOGTIME}$-computable. □

Note that, while using the same *layer technique* as in the NL-completeness proof of reachability in directed acyclic graphs, our reduction cannot be simplified by starting with an acyclic graph: the semigroup is fixed and needs to have this "layered shape" independent of the input graph. Two immediate corollaries of the hardness result are as follows.

Corollary 5.7. *Let $\mathbf{V} \not\subseteq \mathbb{D}\mathbf{S}$ be a variety of finite semigroups. Then, the decision problem* $\textsc{Cayley}$–$\mathrm{SM}(\mathbf{V})$ *is* NL*-complete under* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions.*

Proof. Since, by Theorem 2.11, $B_2 \in \mathbf{V}$, the variety $\mathbf{V}$ contains the variety generated by B_2 which, in turn, contains all finite semigroups satisfying $x^2 = xyx = 0$ by Theorem 2.9. □

Corollary 5.8. *Let* $\mathbf{V}$ *be a variety of finite semigroups (resp. variety of finite monoids) containing N^1. Then, the decision problem* $\textsc{Cayley}$–$\mathrm{SM}(\mathbf{V})$ *is* NL*-complete under* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions.*

Proof. Since $N^1 \in \mathbf{V}$, the variety $\mathbf{V}$ contains the variety generated by N^1 which, by Proposition 2.7, contains all monoids of the form S^1 where S is a finite semigroup which satisfies $x^2 = xyx = 0$. □

Note that the monoid N^1 is not *finite join irreducible*, i.e., there are varieties of finite monoids $\mathbf{V}$ and $\mathbf{W}$ such that $N^1 \notin \mathbf{V}$ and $N^1 \notin \mathbf{W}$ but $N^1 \in \mathbf{V} \vee \mathbf{W}$. This means that there is no unique maximal variety that does not contain N^1. However, there are weaker sufficient conditions for containment of N^1.

Corollary 5.9. *Let* $\mathbf{V}$ *be a variety of finite monoids such that* $\mathbf{V} \not\subseteq \mathbf{Com} \cup \mathbf{CR}$. *Then, the decision problem* $\textsc{Cayley}$–$\mathrm{SM}(\mathbf{V})$ *is* NL*-complete under* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions.*

Proof. If $\mathbf{V} \not\subseteq \mathbf{CR}$, then $\mathbf{V}$ contains the monoid $C^1_{2,1}$. Lemma 4.16 yields $N^1 \in \mathbf{V}$, thus $\textsc{Cayley}$–$\mathrm{SM}(\mathbf{V})$ is NL-complete by Corollary 5.8. □

In view of Corollary 5.4, the previous result now shifts the focus on varieties $\mathbf{V} \subseteq \mathbf{CR}$ with $\mathbf{V} \not\subseteq \mathbf{G} \vee \mathbf{J_1}$. By Theorem 2.11, Theorem 2.12 and Lemma 2.2, the variety $\mathbf{G} \vee \mathbf{J_1}$ contains all varieties of finite monoids $\mathbf{V} \subseteq \mathbf{CR}$ with $U_2 \notin \mathbf{V}$ and $U_2^{\mathrm{op}} \notin \mathbf{V}$. Thus, canonical candidates to investigate further are the varieties $\mathbf{L_1}$ and $\mathbf{R_1}$ which are generated by U_2 and U_2^{op}, respectively. Interestingly enough, we will prove in the next section that the semigroup membership problem for these varieties is in $\mathsf{DLOGTIME}$-uniform AC^0, even though they do not have the polylog CP.

5.1.3 Semigroup Membership in Bands

In this section, we consider the semigroup membership problem for finite idempotent semigroups, so-called *bands*. The semigroup membership problem for idempotent transformation monoids is well-understood. In [BMT92], the following trichotomy theorem was established:

Theorem 5.10 (Beaudry, McKenzie, Thérien)**.** *Let* $\mathbf{V}$ *be a variety of finite monoids with* $\mathbf{V} \subseteq \mathbf{B}$*. Then, exactly one of the following three situations occurs:*

- $\mathbf{V} \subseteq \mathbf{J_1}$ *and* $\textsc{Trans}$–$\mathrm{SM}(\mathbf{V})$ *is in* AC^0*,*
- $\mathbf{V} \not\subseteq \mathbf{J_1}$ *and* $\mathbf{V} \subseteq \mathbf{RB}$ *and* $\textsc{Trans}$–$\mathrm{SM}(\mathbf{V})$ *is* P*-complete,*
- $\mathbf{V} \not\subseteq \mathbf{RB}$ *and* $\textsc{Trans}$–$\mathrm{SM}(\mathbf{V})$ *is* NP*-complete.*

Our first objective is to extend this trichotomy to varieties of finite semigroups. We will show that containment in AC^0 can be extended to the variety $\mathbf{NB}$ which is a strict superset of $\mathbf{J_1}$. Note that Beaudry already provided a NC^2 algorithm for $\textsc{Trans}$–$\mathrm{SM}(\mathbf{NB})$ in his PhD thesis [Bea88b]. For the variety $\mathbf{J_1}$, this was improved to AC^0 in [BMT92]. To lift this result to $\mathbf{NB}$, we need the alternative characterization of normal bands from Proposition 2.5.

Lemma 5.11. $\textsc{Trans}$–$\mathrm{SM}(\mathbf{NB}) \leqslant_{\mathsf{AC}^0} \textsc{Trans}$–$\mathrm{SM}(\mathbf{J_1})$*.*

Proof. For a semigroup $S \in \mathbf{NB}$ and elements $p, s, t, q \in S$, we have $pstq = pstq\,pstq = p(ps)(tq)pstq = p(tq)(ps)pstq = ptq(ps)(pstq)q = ptq(pstq)(ps)q = ptq(ps)(tqps)q = ptq(tqps)(ps)q = ptq\,psq = psq\,ptq$. The last equality uses Proposition 2.5.

Suppose now that we are given a set of transformations X and an additional transformation t on a common finite set such that $S = \langle X \rangle \in \mathbf{NB}$ and we want to decide whether t belongs to S. It is clear that $t \in S$ if and only if there exist $p, x_1, \ldots, x_k, q \in X$ such that $px_1 \cdots x_k q = t$. By the calculation above, this means that $px_1q \cdots px_kq = t$.

This gives rise to the following reduction: for each pair $(p, q) \in X \times X$, we compute the set pXq and use an oracle gate to decide whether t is in the subsemigroup of S generated by pXq. Note that by Proposition 2.5, each of the subsemigroups generated by pXq is a semilattice. We feed the outputs of the oracle gates into a single OR gate to obtain a circuit for the membership problem for S.

Clearly, every entry of each element of each set pXq is computable in $\mathsf{DLOGTIME}$. □

If $\mathbf{V}$ is not contained in $\mathbf{NB}$, there exists a semigroup $S \in \mathbf{V}$ which does not belong to $\mathbf{NB}$. By Proposition 2.5, we obtain that there is a local monoid in S which is not a semilattice. By closure of $\mathbf{V}$ under taking subsemigroups, the variety of all monoids from $\mathbf{V}$ contains a monoid $M \notin \mathbf{J_1}$ and thus, $\textsc{Trans}$–$\mathrm{SM}(\mathbf{V})$ is P-hard. The $\textsc{Trans}$–$\mathrm{SM}(\mathbf{RB})$ algorithm for monoids also works for semigroups and the NP-hardness result carries over to semigroups as well, as already observed in [Bea88b]. Taken all together, we obtain the following trichotomy result.

Theorem 5.12. *Let $\mathbf{V} \subseteq \mathbf{B}$ be a variety of finite semigroups. Then, exactly one of the following three situations occurs:*

- $\mathbf{V} \subseteq \mathbf{NB}$ *and* Trans–SM($\mathbf{V}$) *is in* AC^0*,*
- $\mathbf{V} \not\subseteq \mathbf{NB}$ *and* $\mathbf{V} \subseteq \mathbf{RB}$ *and* Trans–SM($\mathbf{V}$) *is* P*-complete,*
- $\mathbf{V} \not\subseteq \mathbf{RB}$ *and* Trans–SM($\mathbf{V}$) *is* NP*-complete.*

We now proceed to the Cayley semigroup membership problem. In view of the reduction from Proposition 5.1, the previous theorem immediately yields containment of Cayley–SM($\mathbf{NB}$) in AC^0. However, for varieties $\mathbf{V} \subseteq \mathbf{B}$ not contained in $\mathbf{NB}$, this reduction is not very useful: Cayley–SM($\mathbf{V}$) is known to be in NL but Trans–SM($\mathbf{V}$) is P-hard. Nevertheless, for Cayley–SM($\mathbf{RB}$) there is an algorithm which is much more efficient than the generic NL algorithm for Cayley semigroup membership:

Theorem 5.13. Cayley–SM($\mathbf{RB}$) *belongs to* $\Pi_2\mathsf{TIME}(\log n)$*.*

Proof. Suppose we are given a regular band $S \in \mathbf{RB}$, a set $X \subseteq S$ and an element $t \in S$ for which we want to decide whether $t \in \langle X \rangle$.

We will describe an algorithm which can be implemented on a random-access Turing machine with a universal initial state, running in logarithmic time with at most two alternations on each input. The main computation of the algorithm is performed in two parallel universal branches. We will call these branches *left* and *right* branch and we will only describe the computation on the left branch. The procedure in the right branch is exactly the same except for all multiplications being replaced by their left-right dual.

In the left branch, we again branch universally, continuing the computation on two branches, called *base branch* and *continuation branch* of the left branch. In the base branch, we verity that there exists some $x \in X$ such that $xt = t$. In the continuation branch, we, again, branch universally for every element $s \in S$. In each of these $|S|$ branches, we verity that either $st \neq t$ or $ts = s$ or $\exists x \in X \colon (sxs \neq s \wedge sxt = t)$.

Note that the condition $sxs \neq s$ implies $s >_{\mathcal{R}} sx$. If there were $q \in S^1$ with $sxq = s$, then $sxs = sxqxsxq = s(xqxsx)q = s(xqsx)q = s^2 = s$. Here, the third equality uses $S \in \mathbf{RB}$. Thus, whenever the algorithm accepts the input, we can inductively construct a sequence of $k \leqslant |S|$ elements $x_1, \dots, x_k \in X$ such that $x_1 \cdots x_k t = t$ and $tx_1 \cdots x_k = x_1 \cdots x_k$. Analogously, from the right branch, we obtain a sequence of elements $y_1, \dots, y_\ell \in X$ such that $ty_\ell \cdots y_1 = t$ and $y_\ell \cdots y_1 t = y_\ell \cdots y_1$. Together, this yields $x_1 \cdots x_k y_\ell \cdots y_1 = tx_1 \cdots x_k y_\ell \cdots y_1 t = tx_1 \cdots x_k t y_\ell \cdots y_1 t = t$.

Conversely, let us show that if t belongs to the subsemigroup of S generated by X, then the algorithm accepts the input. Suppose that $t = x_1 \cdots x_k$ with $x_1, \dots, x_k \in X$. We only show that the left branch accepts. Acceptance of the right branch follows by symmetry. Clearly, $x_1 t = t$, so the base branch accepts. Now, let $s \in S$ be an arbitrary element with $st = t$ and $ts \neq s$. Choose $i \geqslant 1$ minimal such that $sx_1 \cdots x_i s \neq s$. This means that $sx_1 \cdots x_{i-1} s = s$ and therefore, $sx_i s = sx_1 \cdots x_{i-1} s x_i s = sx_1 \cdots x_i s \neq s$. Similarly, we obtain $sx_i t = sx_i st = sx_1 \cdots x_i s x_1 \cdots x_k = sx_1 \cdots x_k = st = t$. □

We remark that by Corollary 5.5, we know that Cayley–SM(**RB**) does not belong to NPOLYLOGTIME so it seems unlikely that the running time of the algorithm can be improved significantly. However, Theorem 5.13 is certainly much less satisfying than the statement of Theorem 5.12: it does not provide any lower bounds indicating that **RB** is a natural complexity barrier. Of course, it is desirable to prove lower bounds as in the transformation semigroup setting, where we do have hardness results for the membership problem when $\mathbf{V} \not\subseteq \mathbf{RB}$. Unfortunately, we are not aware of *any* non-trivial hardness result, even for Cayley–SM(**B**). Considering the development of complexity results for Cayley–SM(**G**), one should not rule out the possibility that Cayley–SM(**B**) is not hard for any well-known complexity class.

5.2 The Word Problem

We only briefly address the word problem here. It is known that for transformation semigroups, the *uniform word problem* (where the semigroup is given as part of the input) is L-complete under NC^1 reductions [CM87]. It is not difficult to see that this already holds for very restricted inputs, such as unary alphabets. The *non-uniform* variant of the word problem is well-known to be NC^1-complete under non-uniform AC^0 reductions by Barrington's theorem [Bar89]. It is also known that NC^1-hardness already holds whenever the semigroup contains some non-solvable group. For semigroups containing only solvable groups, the non-uniform word problem is in ACC^0 and for aperiodic semigroups, it is in AC^0.

For the uniform word problem for semigroups in Cayley encoding, we immediately obtain membership to L and NC^1-hardness as transfer results. Unfortunately, no stronger bounds are known and this problem is a candidate for having complexity intermediate between NC^1 and L. For *unary alphabets*, we have the following result which is incomparable to the L upper bound but makes it unlikely that this variant of the word problem is hard for any classical complexity class (and actually proves non-hardness under non-uniform qAC^0 reductions). This is in contrast to transformation semigroups, where the uniform word problem for unary alphabets is as hard as in the general case.

Proposition 5.14. *The problem of deciding, given the Cayley table of a semigroup, a word w over a unary alphabet $A = \{a\} \subseteq S$ (in unary encoding) and an element $t \in S$, whether $w = t$ in S is in $\mathsf{DTIME}(\log^2 n)$.*

Proof. A deterministic random-access Turing machine can determine the length ℓ of the input word in logarithmic time using a double binary search. We can then use fast exponentiation to compute a^ℓ in time $\mathcal{O}(\log(\ell)\log(n))$. Here, n denotes the input size. Note that since $\ell \leqslant n$, this yields a total running time in $\mathcal{O}(\log^2 n)$. □

5.3 Language Properties

The most prominent decision problems in language theory are language *emptiness*, *universality*, *inclusion* and *equivalence*. We will consider these problems for languages given as morphisms to finite semigroups. We will also study the *finiteness problem.*

5.3.1 Finite Words

We will mainly consider language problems over finite words and briefly go into extensions to infinite words later. The formal definitions of the *emptiness*, *universality*, *inclusion* and *equivalence* problems are as follows.

Cayley–Emptiness($\mathbf{C}$)

Input: A finite semigroup $S \in \mathbf{C}$, a morphism $h\colon A^+ \to S$ and a set $P \subseteq S$
Question: Is $h^{-1}(P) = \emptyset$?

Cayley–Universality($\mathbf{C}$)

Input: A finite semigroup $S \in \mathbf{C}$, a morphism $h\colon A^+ \to S$ and a set $P \subseteq S$
Question: Is $h^{-1}(P) = A^+$?

Cayley–Inclusion($\mathbf{C}$)

Input: Morphisms $h\colon A^+ \to S$, $g\colon A^+ \to T$ to fin. sg. $S, T \in \mathbf{C}$ and sets $P, Q \subseteq S$
Question: Is $h^{-1}(P) \subseteq g^{-1}(Q)$?

Cayley–Equivalence($\mathbf{C}$)

Input: Morphisms $h\colon A^+ \to S$, $g\colon A^+ \to T$ to fin. sg. $S, T \in \mathbf{C}$ and sets $P, Q \subseteq S$
Question: Is $h^{-1}(P) = g^{-1}(Q)$?

We sometimes omit the class $\mathbf{C}$ to refer to the variants of the problems with $\mathbf{C} = \mathbf{S}$. A subscript $k \in \mathbb{N}$ is used to refer to variants where the size of each accepting set P and Q is at most k. Using Lemma 2.14, it is easy to see that each of the problems is reducible to the corresponding variant for DFAs. Moreover, the following reductions show that the problems above are essentially equivalent under $\mathsf{DLOGTIME}$-uniform AC^0 reductions and also equivalent to the Cayley semigroup membership problem as studied in Section 5.1.

Proposition 5.15. *Let* $\mathbf{C}$ *be a class of finite semigroups and let* $k \in \mathbb{N}$*. Then, the following properties hold:*

1. Cayley–Emptiness($\mathbf{C}$) $\leqslant_{\mathsf{AC}^0}$ Cayley–Emptiness$_1$($\mathbf{C}$).
2. Cayley–Emptiness($\mathbf{C}$) $\leqslant_{\mathsf{AC}^0}$ Cayley–Universality($\mathbf{C}$).
3. Cayley–Emptiness$_k$($\mathbf{C}$) $\leqslant_{\mathsf{AC}^0}$ Cayley–Equivalence$_k$($\mathbf{C}$).
4. Cayley–Universality$_k$($\mathbf{C}$) $\leqslant_{\mathsf{AC}^0}$ Cayley–Equivalence$_k$($\mathbf{C}$).
5. Cayley–Equivalence$_k$($\mathbf{C}$) $\leqslant_{\mathsf{AC}^0}$ Cayley–Inclusion$_k$($\mathbf{C}$).
6. *If the class* $\mathbf{C}$ *is closed under direct products, then* Cayley–Inclusion($\mathbf{C}$) $\leqslant_{\mathsf{AC}^0}$ Cayley–Emptiness($\mathbf{C}$).
7. Cayley–Emptiness$_1$($\mathbf{C}$) $\equiv_{\mathsf{AC}^0}$ Cayley–SM($\mathbf{C}$).

Proof. (1) Let $h\colon A^+ \to S$ be a morphism to a finite semigroup S and let $P \subseteq S$. Then, the language $h^{-1}(P)$ is empty if and only if all languages $h^{-1}(s)$ with $s \in P$ are empty. Using oracle gates for CAYLEY–EMPTINESS$_1(\mathbf{C})$, this can be tested with a DLOGTIME-uniform family of linear-size depth-2 circuits.

(2) Let $h\colon A^+ \to S$ be a morphism to a finite semigroup S and let $P \subseteq S$. Then, the language $h^{-1}(P)$ is empty if and only if $h^{-1}(S \setminus P) = A^+$. To this end, we only need to complement the bit vector representing P.

(3) Let $h\colon A^+ \to S$ be a morphism to a finite semigroup S and let $P \subseteq S$. Then, the language $h^{-1}(P)$ is empty if and only if $h^{-1}(P) = h^{-1}(\emptyset)$. Clearly, this reduction can be performed by a DLOGTIME-uniform family of constant-depth polynomial-size circuits.

(4) Let $h\colon A^+ \to S$ be a morphism to a finite semigroup S and let $P \subseteq S$. Let e be an arbitrary idempotent element of S and let $g\colon A^+ \to S$ be the morphism defined by $g(a) = e$ for all $a \in A$. Then, $g^{-1}(e) = A^+$ and thus, $h^{-1}(P) = A^+$ if and only if $h^{-1}(P) = g^{-1}(e)$. The element e can be computed by a non-deterministic random-access Turing machine in logarithmic time. Thus, the reduction can be carried out by a DLOGTIME-uniform family of constant-depth polynomial-size circuits.

(5) This follows immediately from the fact that two languages $K, L \subseteq A^+$ are equal if and only if $K \subseteq L$ and $L \subseteq K$. This test can be conducted by a circuit with two oracle gates and an additional AND gate.

(6) Let $h\colon A^+ \to S$ and $g\colon A^+ \to T$ be morphisms to finite semigroups S and T, let $P \subseteq S$ and let $Q \subseteq T$. Then, $h^{-1}(P) \subseteq g^{-1}(Q)$ if and only if $h^{-1}(P) \setminus g^{-1}(Q) = \emptyset$. We compute the multiplication table for $S \times T$, the morphism $f\colon A^+ \to S \times T$ defined by $f(a) = (h(a), g(a))$ and the set $R = P \times (T \setminus Q)$. By construction, $f^{-1}(R) = h^{-1}(P) \setminus g^{-1}(Q)$. It is straightforward to verify that the reduction is computable by a DLOGTIME-uniform family of constant-depth polynomial-size circuits. Note that the elements of $S \times T$ can be represented as concatenations of the binary representations of their components. Rows and columns with indices that do not correspond to such a concatenation are filled with the padding value.

(7) Let $h\colon A^+ \to S$ and $s \in S$. Then $h^{-1}(s) \neq \emptyset$ if and only if s belongs to the subsemigroup of S generated by $\{h(a) \mid a \in A\}$. Conversely, let S be a semigroup, let $X \subseteq S$ and $t \in S$. Let $h\colon X^+ \to S$ be the *evaluation morphism* defined by $h(x) = x$ for all $x \in X$. Then t belongs to the subsemigroup of S generated by X if and only if $h^{-1}(t) \neq \emptyset$. Both reductions are trivial since we can interpret morphisms as generating sets and vice versa (the order of the letters does not matter in this case). □

The proposition also shows that for classes closed under direct products, language emptiness, inclusion and equivalence are equivalent to the variants with singleton accepting sets under DLOGTIME-uniform AC^0 reductions. Only the universality problem becomes easier if the size of the accepting set is fixed.

Proposition 5.16. *For every $k \in \mathbb{N}$, the decision problem* CAYLEY–UNIVERSALITY$_k$ *belongs to* coNLOGTIME.

Proof. Let $h\colon A^+ \to S$ be a morphism to a finite semigroup S and let $P \subseteq S$ with $|P| \leqslant k$. Suppose that $h^{-1}(P) \neq A^+$, i.e., there exists $u \in A^+$ with $h(u) \notin P$. We may choose $u = a_1 \cdots a_\ell$ to be a word of minimal length with this property. This means that $h(a_1 \cdots a_i) \in P$ for $1 \leqslant i < \ell$ and $h(a_1 \cdots a_i) \neq h(a_1 \cdots a_j)$ for $1 \leqslant i < j \leqslant \ell$. Since for $i \in \{1, \ldots, \ell - 1\}$, the prefixes $h(a_1 \cdots a_i)$ are pairwise disjoint elements of P, this yields $\ell \leqslant |P| + 1 \leqslant k + 1$. Thus, $h^{-1}(P) \neq A^+$ if and only if there exists a word of length at most $k+1$ which is not mapped to an element of P under h.

We can guess this word letter by letter, computing the image of the currently guessed prefix at the same time. As soon as $k+1$ letters are guessed (or the machine nondeterministically decides to stop earlier), we check that the computed element does not belong to the set P. Guessing a single letter, computing the image of a letter under h and performing a single multiplication can be done in time $\mathcal{O}(\log n)$. Since k is a constant, the total running time is in $\mathcal{O}(\log n)$ as well. □

Another natural decision problem for languages is the *finiteness problem* which asks whether a given language contains only finitely many pairwise disjoint elements. It is formally defined as follows.

CAYLEY–FINITENESS($\mathbf{C}$)	
Input:	A finite semigroup $S \in \mathbf{C}$, a morphism $h\colon A^+ \to S$ and a set $P \subseteq S$
Question:	Is $\lvert h^{-1}(P) \rvert < \infty$?

Our complexity results for the finiteness problem are based on the following lemma.

Lemma 5.17. *Let $h\colon A^+ \to S$ be a morphism to a finite semigroup and let $P \subseteq S$. Then, the language $h^{-1}(P)$ is infinite if and only if there exist $p, e, q \in h(A^+)$ such that $e^2 = e$ and $peq \in P$.*

Proof. Suppose that $h^{-1}(P)$ is infinite. Then, there exists some word $w \in A^+$ such that $h(w) \in P$ and $|w| > |S|$. By Lemma 2.1, we can factorize $h(w) = peq$ with $p, e, q \in h(A^+)$ and with $e^2 = e$.

Conversely, let $p, e, q \in h(A^+)$ with $e^2 = e$ and $peq \in P$. Let $u, v, w \in A^+$ such that $h(u) = p$, $h(v) = e$ and $h(w) = q$. Then, for each $i \geqslant 1$, we have $h(uv^iw) = pe^iq = peq \in P$, which suffices to show that $h^{-1}(P)$ is infinite. □

There is a simple reduction of the finiteness problem to the emptiness problem with a singleton accepting set.

Proposition 5.18. *Let $\mathbf{C}$ be a class of finite semigroups. Then, the decision problem* CAYLEY–FINITENESS($\mathbf{C}$) *is reducible to* CAYLEY–EMPTINESS$_1$($\mathbf{C}$) *via* DLOGTIME-*uniform* AC^0 *reductions.*

Proof. By Lemma 5.17, it suffices to check for each $(p, e, q) \in S$ whether $p, e, q \in h(A^+)$ and $e^2 = e$ and $peq \in P$. If we make the triple (p, e, q) part of the input, this can be verified by a deterministic logarithmic-time random-access Turing machine with an oracle for CAYLEY–EMPTINESS$_1$($\mathbf{C}$). □

For classes of finite monoids, there is an even closer relationship between the finiteness problem and the emptiness problem as stated in the following result.

Proposition 5.19. *Let* $\mathbf{C}$ *be a class of finite monoids. Then, the decision problems* Cayley–Finiteness($\mathbf{C}$) *and* Cayley–Emptiness($\mathbf{C}$) *are equivalent via* $\mathsf{DLOGTIME}$-*uniform* AC^0 *reductions.*

Proof. By Proposition 5.18, it suffices to describe a reduction from the emptiness problem to finiteness. Suppose we are given a recognizing morphism $h\colon A^+ \to M$ and an accepting set $P \subseteq M$ for which we want to decide whether $h^{-1}(P)$ is empty.

We first test whether the neutral element 1 belongs to $h(A^+)$. This check can be performed in $\mathsf{NLOGTIME}$ since $1 \in h(A^+)$ if and only if there exists some $a \in A$ with $h(a) \mathrel{\mathcal{J}} 1$, whence $h(a^\omega) = 1$. If 1 belongs to $h(A^+) \cap P$, we reject the input because in this case, the language $h^{-1}(P) \supseteq h^{-1}(1)$ is clearly infinite. Otherwise, we may remove 1 from the accepting set P without affecting emptiness of the language $h^{-1}(P)$. We then extend the alphabet A by adding a fresh letter c which is mapped to the neutral element 1 of M under h.

It is clear that any word accepted by this new recognizing morphism can be converted into a word recognized by the old morphism by removing all occurrences of the letter c. The resulting word is non-empty since $1 \notin h^{-1}(P)$. Conversely, if the new morphism recognizes some word w, it also recognizes all words wc^i with $i \geqslant 1$. Thus, the new language is infinite whenever the original language is non-empty, and empty otherwise.

The neutral element is the unique element e satisfying $\forall x\colon (ex = x \wedge xe = x)$. Thus, we can compute this element by a $\Sigma_2\mathsf{TIME}(\log n)$-Turing machine. The question of whether $1 \in h(A^+)$ is decidable in $\mathsf{NLOGTIME}$ as described above. It follows that the reduction can be performed by a $\mathsf{DLOGTIME}$-uniform family of AC^0 circuits. □

The previous result can not be extended to arbitrary classes of finite semigroups. For example, efficient algorithms for the emptiness problem for groups can be lifted to obtain efficient algorithms for the finiteness problem for nilpotent extensions.

Proposition 5.20. *Let* $\mathbf{H}$ *be a variety of finite groups. Then, the decision problem* Cayley–Finiteness($\mathbf{H}^{\mathbf{N}}$) *is equivalent to* Cayley–Emptiness($\mathbf{H}$) *via* $\mathsf{DLOGTIME}$-*uniform* AC^0 *reductions.*

Proof. Every language L recognized by a finite group G is either empty or infinite: if $w \in L$, then $w^{i|G|+1} \in L$ for all $i \in \mathbb{N}$. Thus, the reduction of Cayley–Emptiness($\mathbf{H}$) to Cayley–Finiteness($\mathbf{H}$) is trivial. Note that $\mathbf{H} \subseteq \mathbf{H}^{\mathbf{N}}$.

Conversely, let $h\colon A^+ \to S$ be a morphism to a finite semigroup $S \in \mathbf{H}^{\mathbf{N}}$ and let $P \subseteq S$. Note that since $S \in \mathbf{G}^{\mathbf{N}}$, there is a unique idempotent element $e \in S$ and we have $h(a^\omega) = e$ for each $a \in A$. We can guess this idempotent element (and verify that it is indeed idempotent) in $\mathsf{NLOGTIME}$. We then define a new morphism $g\colon A^+ \to S$ by $g(a) = h(a)e$. Since every element $g(a)$ for $a \in A$ can be computed in $\mathsf{NLOGTIME}$, the morphism g can be constructed from h by a $\mathsf{DLOGTIME}$-uniform family of AC^0 circuits. Moreover, by definition, $g(A) \subseteq SE(S)S$ and therefore, $g(A^+) \in \mathbf{H}$. The claim is that $g^{-1}(P)$ is non-empty if and only if $h^{-1}(P)$ is infinite.

We will use the fact that the unique idempotent element e of S is central. This is easily verified by the equalities $es = s^\omega s = ss^\omega = se$ which hold for all $s \in S$.

If the language $g^{-1}(P)$ is non-empty, there exists some word $w = a_1 \cdots a_\ell$ with $g(w) = h(a_1)e \cdots h(a_\ell)e = h(w)e \in P$. But then, all words of the form wa^i, with i being a multiple of ω, are mapped to $h(w)e$ under h, so $h^{-1}(P)$ is infinite.

If $h^{-1}(P)$ is infinite, there exists a word $w = a_1 \cdots a_\ell$ with $h(w) \in P$ and with $\ell \geqslant |S|$. Together with the fact that the idempotent element is unique and central, Lemma 2.1 yields $h(w) = h(a_1)e \cdots h(a_\ell)e = g(w)$. □

Note that $\mathbf{I}^\mathbf{N} = \mathbf{N}$, so Cayley–Finiteness($\mathbf{N}$) belongs to DLOGTIME-uniform AC^0 by the previous result. On the other hand, Cayley–Emptiness($\mathbf{N}$) is NL-complete by Theorem 5.6 and Proposition 5.15. Another direct consequence is membership of Cayley–Finiteness($\mathbf{G}^\mathbf{N}$) to $\mathsf{L} \cap \mathsf{NPOLYLOGTIME} \cap \mathsf{FOLL}$.

5.3.2 Infinite Words

In model checking, one often considers languages over *infinite words*. An infinite word over some alphabet A is an infinite sequence $a_1 a_2 \cdots$ of letters of A. Such a sequence can also be viewed as a mapping $\mathbb{N} \to A$. We therefore denote the set of all infinite words over A by $A^\mathbb{N}$.

One can easily extend the notion of recognition by semigroups to infinite words. Let S be a finite semigroup and let $h\colon A^+ \to S$ be a morphism. A pair (s, e) of elements of S is called *linked pair* if $se = s$ and $e^2 = e$. Let (s, e) be a linked pair of S. The language *weakly recognized by h via (s, e)* consists of all infinite words from $A^\mathbb{N}$ which can be factorized as $uv_0v_1 \cdots$ such that $h(u) = s$ and $h(v_i) = e$ for $i \geqslant 0$. By extension, for a set of linked pairs P, the language weakly recognized by h via P is the union of all languages weakly recognized by h via the elements of P. A language $L \subseteq A^\mathbb{N}$ is weakly recognized by h if it is weakly recognized via some set of linked pairs of S.

Two linked pairs (s, e) and (t, f) are *conjugate* if there exist $x, y \in S$ such that $sx = t$, $xy = e$ and $yx = f$. It is easy to verify that conjugation is an equivalence relation on the set of linked pairs. A language is *(strongly) recognized* by a morphism h if it is weakly recognized by h via a set of linked pairs P which is closed under conjugation.

We are mainly interested in strong recognition. Strongly recognizing morphisms behave nicely with regard to common closure operations. On weakly recognizing morphisms, even complementation involves an inherent blow-up. This is similar to the situation of non-deterministic automata over finite words. We refer to [FK16] for an overview on the descriptional complexity of weakly recognizing morphisms.

Note that if a language $L \subseteq A^\mathbb{N}$ is weakly recognized by a morphism h via some set P and, additionally, L is strongly recognized by h, then L is also weakly recognized by the closure of P under conjugation. One can assume that accepting sets provided alongside strongly recognizing morphisms are always closed under conjugation.

Lemma 5.21. *Suppose a finite semigroup S is given in Cayley encoding. The problem of deciding whether a pair $(s, t) \in S \times S$ is a linked pair is in* DLOGTIME. *The problem of deciding whether two linked pairs (s, e) and (t, f) are conjugate is in* NLOGTIME.

Proof. Testing whether $st = s$ and $t^2 = t$ only requires two table lookups of $\mathcal{O}(\log n)$ bits. To test whether two linked pairs (s, e) and (t, f) are conjugate, we guess elements $x, y \in S$ in time $\mathcal{O}(\log n)$ and verify that $sx = t$, $xy = e$, $yx = f$ by performing three table lookups of $\mathcal{O}(\log n)$ bits. □

Lemma 5.22. *Given a finite semigroup S in Cayley encoding and a set of linked pairs P of S, the closure of P under conjugation can be computed in* $\mathsf{DLOGTIME}$*-uniform* AC^0.

Proof. The statement immediately follows from the previous lemma, the fact that $\mathsf{DLOGTIME} \subseteq \mathsf{NLOGTIME} \subseteq \mathsf{LH}$ and the fact that there are only $|S|^2$ pairs in $S \times S$. □

It is straightforward to adapt the reductions between language emptiness, universality, inclusion and equivalence from Proposition 5.15 to strongly recognizing morphisms. Assuming that P is closed under conjugation, language complementation corresponds to taking the relative complement of P with respect to the set of all linked pairs. This is required to show that language emptiness is reducible to universality. The direct product construction for the reduction of language inclusion to emptiness works as in the case of finite words (again, assuming that the accepting sets are closed under conjugation). However, the equivalence with Cayley semigroup membership does not hold. It may seem surprising that the emptiness problem over infinite words is actually *easier* than the emptiness problem over finite words.

Proposition 5.23. *For a variety of finite semigroups* $\mathbf{V}$*, the emptiness problem for recognizing morphisms over infinite words is reducible to* Cayley–Finiteness($\mathbf{V}$) *via* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions.*

Proof. Let $h\colon A^+ \to S$ be a morphism to a finite semigroup S and let (s, e) be a linked pair of S. The language weakly recognized by h via (s, e) is non-empty if and only if both $h^{-1}(s)$ and $h^{-1}(e)$ are non-empty. We claim that this is, in turn, equivalent to both $h^{-1}(s)$ and $h^{-1}(e)$ being infinite. To see this, note that if $u, v \in A^+$ are words with $h(u) = s$ and $h(v) = e$, then $h(uv^i) = se = s$ and $h(v^i) = e$ for all $i \geqslant 1$. □

In particular, by Proposition 5.20, the emptiness problem over infinite words for the variety $\mathbf{G^N}$ is in $\mathsf{NPOLYLOGTIME}$ whereas the emptiness problem over finite words for this variety is NL-complete. However, for varieties of finite monoids, the emptiness problems over infinite words and finite words are equivalent under $\mathsf{DLOGTIME}$-uniform AC^0 reductions.

Proposition 5.24. *For a variety of finite monoids* $\mathbf{V}$*, the emptiness problem for recognizing morphisms over infinite words is equivalent to* Cayley–Emptiness($\mathbf{V}$) *via* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions.*

Proof. The reduction from left to right follows from Proposition 5.23 and Proposition 5.18. For the converse direction, let $h\colon A^+ \to M$ be a morphism to a finite monoid M and let $m \in M$. We may add a new letter c to A which is mapped to the neutral element 1 under h without affecting (non-)emptiness of the set $h^{-1}(m)$; the case $m = 1$ is handled as in Proposition 5.19. Then, clearly, $h^{-1}(m)$ is non-empty if and only if the language weakly recognized by h via the linked pair $(m, 1)$ is non-empty. □

5.4 Intersection Non-Emptiness

In 1977, Kozen showed that the problem of deciding non-emptiness of the intersection of the languages recognized by a set of given DFAs is PSPACE-complete [Koz77]. This result has been the building block for numerous hardness results in formal language theory and related fields; see e.g. [Ber97, CH91, DGH05, JR91]. Various special cases, such as bounding the number of automata or the number of accepting states, were investigated in follow-up work; see [HK11] for a survey. Another type of natural restrictions is requiring the automata in the input to have certain structural properties. These properties are often expressed in terms of membership to a certain variety of finite semigroups; in the automaton setting, one considers the transformation semigroups of the automata. For example, it is known that PSPACE-completeness already holds for any variety of finite semigroups not contained within $\mathbb{LDS}$ whereas the restriction of the intersection non-emptiness problem to $\mathcal{R}$-trivial semigroups is in NP. Formally, we define the intersection non-emptiness problem for a class of finite semigroups $\mathbf{C}$ as follows:

DFA–Intersection($\mathbf{C}$)	
Input:	DFAs $A_1, \ldots, A_k$ with transition semigroups from $\mathbf{C}$
Question:	Is $L(A_1) \cap \cdots \cap L(A_k) \neq \emptyset$?

Moreover, we sometimes use DFA–Intersection$_m(\mathbf{C})$ to denote the restriction of DFA–Intersection($\mathbf{C}$) where each of the automata is allowed to have at most m accepting states.

In [FK18c], Kufleitner and the author studied the problem for finite monoids and in [Fle18b], the author studied the problem for finite semigroups. For semigroups, the problem is formally defined as follows.

Cayley–Intersection($\mathbf{C}$)	
Input:	Finite semigroups $S_1, \ldots, S_k \in \mathbf{C}$, morphisms $h_i \colon A^+ \to S_i$, sets $P_i \subseteq S_i$
Question:	Is $h_1^{-1}(P_1) \cap \cdots \cap h_k^{-1}(P_k) \neq \emptyset$?

The restriction of Cayley–Intersection($\mathbf{C}$), where each of the accepting sets is allowed to have at most m elements, is denoted by Cayley–Intersection$_m(\mathbf{C})$.

As for the previously discussed problems, there is a straightforward reduction from the Cayley semigroup setting to the automaton setting. Moreover, the intersection non-emptiness problem is closely linked to the membership problem for transformation semigroups. This connection is captured in the following proposition.

Proposition 5.25. *Suppose that* $\mathbf{C}$ *is an arbitrary class of finite semigroups. Then, we have* Cayley–Intersection$_m(\mathbf{C}) \leqslant_{\mathsf{AC}^0}$ DFA–Intersection$_m(\mathbf{C})$ *and, likewise,* Trans–SM($\mathbf{C}$) $\leqslant_{\mathsf{AC}^0}$ DFA–Intersection$_1(\mathbf{C})$. *Moreover, if* $\mathbf{C}$ *is closed under taking finite direct products, then* Cayley–Intersection$_1(\mathbf{C}) \leqslant_{\mathsf{AC}^0}$ Trans–SM($\mathbf{C}$).

Proof. The first part of the statement follows immediately from Lemma 2.14.

For Trans–SM($\mathbf{C}$) $\leqslant_{\mathsf{AC}^0}$ DFA–Intersection$_1(\mathbf{C})$, suppose that we are given transformations $x_1, \ldots, x_k \colon Q \to Q$ and an additional transformation $t \colon Q \to Q$. We construct $|Q|$ deterministic finite automata A_q (with $q \in Q$). These automata all use the same alphabet $\{x_1, \dots, x_k\}$, the same set of states Q and the same transitions, induced by the transformations $x_1, \dots, x_k$. The initial state of A_q is q and the unique final state is $t(q)$. By construction, the intersection of the languages recognized by these automata is non-empty if and only if t is in the subsemigroup $\langle x_1, \dots, x_k \rangle$ of the full transformation semigroup on Q.

For Cayley–Intersection$_1(\mathbf{C}) \leqslant_{\mathsf{AC}^0}$ Trans–SM$(\mathbf{C})$, we perform the conversion of semigroups to DFAs on all semigroups in parallel: if the input consists of finite semigroups $S_1, \dots, S_k$, corresponding morphisms $h_i \colon A^+ \to S_i$ and accepting sets $P_i = \{t_i\}$, we define a set $Q = S_1^1 \cup \dots \cup S_k^1$ and a set of transformations $f_a \colon Q \to Q$ by $f_a(s) = s \cdot h_i(a)$ for each $i \in \{1, \dots, k\}$ and $s \in S_i^1$. An additional transformation $t \colon Q \to Q$ is defined by $t(s) = s \cdot t_i$ for $i \in \{1, \dots, k\}$ and $s \in S_i^1$. It is easy to verify that t belongs to the subsemigroup of the full transformation semigroup on Q generated by $\{f_a \mid a \in A\}$ if and only if the intersection $h_1^{-1}(t_1) \cap \dots \cap h_k^{-1}(t_k)$ is non-empty. It is also easy to verify that this subsemigroup is isomorphic to the subsemigroup of the direct product $S_1 \times \dots \times S_k$ generated by the elements $(h_1(a), \dots, h_k(a))$ with $a \in A$.

The reductions can be performed by a DLOGTIME-uniform family of AC^0 circuits by arguments similar to those given in the proof of Lemma 2.14. □

Since one can guess the elements of an accepting set reached by a witness for intersection non-emptiness, we also have the following proposition.

Proposition 5.26. *The decision problem* Cayley–Intersection$(\mathbf{C})$ *belongs to* NP *if and only if* Cayley–Intersection$_1(\mathbf{C})$ *belongs to* NP.

In fact, the slightly stronger statement of Cayley–Intersection$(\mathbf{C})$ being NP-*reducible* to Cayley–Intersection$_1(\mathbf{C})$ holds. Since we do not need this stronger result, we refrain from formally introducing the notions of NP transducers and NP reducibility.

For automata, one can use the same idea and guess the action of a witness for intersection non-emptiness on each state to obtain a reduction from intersection non-emptiness to the membership problem in transformation semigroups.

Proposition 5.27. *The decision problem* DFA–Intersection$(\mathbf{C})$ *belongs to* NP *if and only if* Trans–SM$(\mathbf{C})$ *belongs to* NP.

The previous results show that above NP, the difficulty of the intersection non-emptiness problem narrows down to transformation semigroup membership for a given variety. In particular, while DFA–Intersection$(\mathbf{R})$ belongs to NP, we do not know whether or not DFA–Intersection$(\mathbf{L})$ is in NP.

Interestingly, this difficulty is easily eliminated in the algebraic setting. By Proposition 5.25, we know that Cayley–Intersection$(\mathbf{R})$ is in NP which immediately yields membership of Cayley–Intersection$(\mathbf{L})$ to NP by symmetry: the conversion of Cayley encodings of semigroups to encodings for the corresponding opposite semigroups can be performed in DLOGTIME-uniform AC^0.

5.4.1 Efficient Algorithms

We recall that circuits properties are closely linked to efficient algorithms for the Cayley semigroup membership problem. Similarly, product circuits properties are closely linked to efficient algorithms for the intersection non-emptiness problem.

Theorem 5.28. *Let* $\mathbf{C}$ *be a class of finite semigroups with the* $f(n)$ *product circuits property. Then* $\textsc{Cayley–Intersection}(\mathbf{C})$ *belongs to* $\Sigma_2\mathsf{TIME}((f(n))^2 \log n)$.

Proof. Suppose that our input consists of k morphisms $h_i \colon A^+ \to S_i$ and accepting sets $P_i \subseteq S_i$ with $1 \leqslant i \leqslant k$. We first guess an admissible encoding of a Cayley circuit of size at most $f(n)$ as in the proof of Theorem 3.15. We also guess a sequence $a_1, \ldots, a_m$ of at most $f(n)$ letters of A on a separate work tape. We then branch universally, creating a separate branch for each morphism $i \in \{1, \ldots, k\}$. The number of the active morphism is remembered on a separate work tape. In the branch for the i-th morphism, we first replace each letter $a \in A$ in the sequence of letters $a_1, \ldots, a_m$ by the corresponding image $h_i(a)$. We then check whether the encoded Cayley circuit computes some element from P_i on the input sequence $(h_i(a_1), \ldots, h_i(a_m))$ as described in the proof of Theorem 3.14.

Guessing both the SLP and the letters corresponding to the elements assigned to the input gates before branching universally guarantees that computation is "synchronized" between the individual morphisms, i.e., the resulting SLP indeed yields a witness for intersection non-emptiness; see Proposition 4.2 for details. □

Corollary 5.29. *Let* $\mathbf{C}$ *be a class of finite semigroups. If* $\mathbf{C}$ *has the* const *PCP, then* $\textsc{Cayley–Intersection}(\mathbf{C})$ *belongs to* $\Sigma_2\mathsf{TIME}(\log n) \subseteq \mathsf{AC}^0$. *If* $\mathbf{C}$ *has the* polylog *PCP, then* $\textsc{Cayley–Intersection}(\mathbf{C})$ *belongs to* $\mathsf{PolyLH} \subseteq \mathsf{qAC}^0$.

Proposition 4.5 and Theorem 4.24 yield the following additional corollaries.

Corollary 5.30. *For each* $k \in \mathbb{N}$, *the decision problem* $\textsc{Cayley–Intersection}(\mathbb{LI}_k)$ *belongs to* $\Sigma_2\mathsf{TIME}(\log n) \subseteq \mathsf{AC}^0$.

Corollary 5.31. *Let* $\mathbf{C} \subseteq \mathbf{Com} \cap \mathbb{LI}$ *be a class of finite semigroups of bounded index. Then,* $\textsc{Cayley–Intersection}(\mathbf{C})$ *belongs to* $\mathsf{PolyLH} \subseteq \mathsf{qAC}^0$.

Given a class of finite semigroups $\mathbf{C}$ with the poly PCP, Theorem 5.28 yields a $\Sigma_2\mathsf{TIME}(n^{\mathcal{O}(1)})$ algorithm for $\textsc{Cayley–Intersection}(\mathbf{C})$. A better algorithm is obtained by transforming the universal verification step into sequential verification which immediately yields the following result.

Theorem 5.32. *Let* $\mathbf{C}$ *be a class of finite semigroups with the* $f(n)$ *product circuits property. Then* $\textsc{Cayley–Intersection}(\mathbf{C})$ *belongs to* $\mathsf{NTIME}(n \cdot (f(n))^2 \log n)$.

Corollary 5.33. *If* $\mathbf{C}$ *has the* poly *PCP, then* $\textsc{Cayley–Intersection}(\mathbf{C})$ *is in* NP.

This immediately yields NP algorithms for the varieties $\mathbf{G} \vee \mathbf{J_1}$, $\mathbf{Com}$, $\mathbf{R}$, $\mathbf{L}$ and $\mathbb{LI}$. We will later see that intersection non-emptiness is NP-complete for each of these varieties. For further results on efficient algorithms, we refer to [FK18c].

5.4.2 Hardness Results

In this section, we generalize Kozen's PSPACE-hardness result for intersection non-emptiness, and then prove NP-hardness for very restricted classes of finite semigroups, thereby complementing the NP algorithms given in the previous subsection.

The PSPACE-hardness result is generalized in two ways. Firstly, we show that hardness already holds in the semigroup setting with singleton accepting sets. By Proposition 5.25, hardness of the membership problem in transformation semigroups and hardness of intersection non-emptiness for automata follow immediately. Secondly, we prove that hardness even holds when the only semigroup allowed in the input is B_2^1. We recall that classes containing B_2^1 do not have the poly PCP as shown in Proposition 4.22. To simplify the main proof, we will first show two technical lemmas. These lemmas describe classes of languages which are recognized by the semigroup B_2^1.

Lemma 5.34. *Let A be a finite alphabet and let B, C, D, E, F be (possibly empty) pairwise disjoint subsets of A. Then, each of the languages $E^*B(D \cup E)^*$, $(D \cup E)^*CE^*$ and $(E^*B(E \cup F)^*CE^* \cup E^*DE^*)^+$ is the preimage of an element of a morphism $h\colon A^+ \to B_2^1$.*

Proof. For $E^*B(D \cup E)^*$, consider the morphism $h\colon A^+ \to B_2^1$ defined by

$$h(c) = \begin{cases} 1 & \text{if } c \in E, \\ b & \text{if } c \in B, \\ ab & \text{if } c \in D, \\ 0 & \text{if } c \in A \setminus (B \cup D \cup E). \end{cases}$$

By construction, we have $h^{-1}(b) = E^*B(D \cup E)^*$. For $(D \cup E)^*CE^*$, one can use a symmetrical construction.

For $(E^*B(E \cup F)^*CE^* \cup E^*DE^*)^+$, we define $h\colon A^+ \to B_2^1$ by

$$h(c) = \begin{cases} 1 & \text{if } c \in E, \\ a & \text{if } c \in B, \\ b & \text{if } c \in C, \\ ab & \text{if } c \in D, \\ ba & \text{if } c \in F, \\ 0 & \text{if } c \in A \setminus (B \cup C \cup D \cup E \cup F). \end{cases}$$

The preimage of ab is the desired language. □

Lemma 5.35. *Let A be a finite alphabet, let $n \in \mathbb{N}$ and let $A_1, \dots, A_n$ be pairwise disjoint subsets of A. Then the language $(A_1 \cdots A_n)^+$ can be written as an intersection of n languages, each of which is the preimage of an element of a morphism $h\colon A^+ \to B_2^1$.*

Proof. Let $B = A_1 \cup \dots \cup A_n$. For each $i \in \{1, \dots, n-1\}$, we define an alphabet $D_i = B \setminus (A_i \cup A_{i+1})$ and a language $L_i = (A_iA_{i+1} \cup D_i)^+$. We also let $L_n = (A_1D_n^*A_n)^+$

with $D_n = B \setminus (A_1 \cup A_n)$. By construction, we have $L_1 \cap \cdots \cap L_n = (A_1 \cdots A_n)^+$ and by Lemma 5.34, each of the languages L_i is the preimage of an element under a morphism to B_2^1. □

Essentially, the upcoming PSPACE-hardness proof can be viewed as a *master reduction* but to reduce the amount of technical details, we choose a presentation based on tiling systems. A *tiling system* is a tuple $\mathcal{T} = (\Lambda, T, n, f, b)$ where Λ is a finite set of *labels*, $T \subseteq \Lambda \times \Lambda \times \Lambda \times \Lambda$ are the so-called *tiles*, $n \in \mathbb{N}$ is the *width* and $f, b \in T^n$ are the *first row* and *bottom row*. For a tile $t = (t_w, t_e, t_s, t_n) \in T$, we let $\lambda_w(t) = t_w$, $\lambda_e(t) = t_e$, $\lambda_s(t) = t_s$ and $\lambda_n(t) = t_n$. These labels can be thought of as labels in *west*, *east*, *south* and *north* direction. A *corridor tiling* of $\mathcal{T}$ is a mapping $\tau\colon \{1, \dots, m\} \times \{1, \dots, n\} \to T$ with $m \in \mathbb{N} \setminus \{0\}$ such that the following properties hold:

1. $\tau(1,1)\tau(1,2) \cdots \tau(1,n) = f$,
2. $\lambda_e(\tau(i,j)) = \lambda_w(\tau(i,j+1))$ for $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant n-1$,
3. $\lambda_s(\tau(i,j)) = \lambda_n(\tau(i+1,j))$ for $1 \leqslant i \leqslant m-1$ and $1 \leqslant j \leqslant n$,
4. $\tau(m,1)\tau(m,2) \cdots \tau(m,n) = b$.

The *corridor tiling problem* asks for a given tiling system $\mathcal{T}$ whether there exists a corridor tiling of $\mathcal{T}$. It is well-known that the corridor tiling problem is PSPACE-complete [Chl86], allowing us to prove the first main theorem of this section. The proof is essentially the same as in [FK18c] but our statement is stronger.

Theorem 5.36. Cayley–Intersection$_1(\{B_2^1\})$ *is* PSPACE*-complete.*

Proof. Let $\mathcal{T} = (\Lambda, T, n, f, b)$ be a tiling system. The objective is to construct a language L which is non-empty if and only if there exists a valid corridor tiling of $\mathcal{T}$.

We may assume without loss of generality that $\lambda_w(t) \neq \lambda_e(t)$ and $\lambda_s(t) \neq \lambda_n(t)$ for all tiles $t \in T$. If, for example, $\lambda_w(t) = \lambda_e(t)$ for a tile $t \in T$, we create a copy μ' of the label $\mu = \lambda_w(t) = \lambda_e(t)$ and replace every tile t' with $\lambda_w(t') = \mu$ by two copies. In one of these copies, we replace the west label with μ'. We repeat this for all other directions and finally remove all tiles with $\lambda_w(t) = \lambda_e(t) \in \{\mu, \mu'\}$.

We define an alphabet $A = T \times \{0,1,2\} \times \{1, \dots, n\}$. Intuitively, the letters of A correspond to positions in a tiling. The first component describes the tile itself, the second component specifies whether the tile is in the first row, some intermediate row or in the bottom row and the third component specifies the column. For each $j \in \{1, \dots, n\}$ and $\mu \in \Lambda$, let $C_j = T \times \{0,1,2\} \times \{j\}$ and $D_j = A \setminus C_j$ and

$$\begin{aligned}
W_\mu &= \{(t,i,j) \in A \mid \lambda_w(t) = \mu, j > 1\}, & N_{j,\mu} &= \{(t,i,j) \in A \mid \lambda_n(t) = \mu, i > 0\}, \\
E_\mu &= \{(t,i,j) \in A \mid \lambda_e(t) = \mu, j < n\}, & S_{j,\mu} &= \{(t,i,j) \in A \mid \lambda_s(t) = \mu, i < 2\}, \\
X_\mu &= A \setminus (W_\mu \cup E_\mu), & Y_{j,\mu} &= C_j \setminus (N_{j,\mu} \cup S_{j,\mu}).
\end{aligned}$$

Note that by the assumption above, $W_\mu \cap E_\mu = \emptyset$ and $N_{j,\mu} \cap S_{j,\mu} = \emptyset$ for each $\mu \in \Lambda$ and for $1 \leqslant j \leqslant n$. Let $F_j = \{(t_j, 0, j)\}$ and $B_j = \{(u_j, 2, j)\}$ where t_j and u_j are the tiles uniquely determined by $f = t_1 \cdots t_n$ and $b = u_1 \cdots u_n$. Let $\overline{F}_j = \{(t,i,j) \in A \mid i > 0\}$ and $\overline{B}_j = \{(t,i,j) \in A \mid i < 2\}$. We define

$$K = \bigcap_{1 \leqslant j \leqslant n} D_j^* F_j (\overline{F}_j \cup D_j)^* \;\cap \bigcap_{1 \leqslant j \leqslant n} (\overline{B}_j \cup D_j)^* B_j D_j^* \;\cap \bigcap_{\mu \in \Lambda} (E_\mu W_\mu \cup X_\mu)^+$$
$$\cap \bigcap_{\substack{\mu \in \Lambda, \\ 1 \leqslant j \leqslant n}} (D_j^* S_{j,\mu} D_j^* N_{j,\mu} D_j^* \cup D_j^* Y_{j,\mu} D_j^*)^+$$

and $L = (C_1 \cdots C_n)^+ \cap K$. By Lemma 5.34 and Lemma 5.35, the language L can be represented by a CAYLEY–INTERSECTION$_1(\{B_2^1\})$ instance with polynomially many morphisms to B_2^1 and with singleton accepting sets. Moreover, it is not difficult to see that this instance can be constructed by a log-space transducer on input $\mathcal{T}$. □

By the exclusion characterizations of $\mathbb{D}\mathbf{S}$ and $\mathbb{L}\mathbb{D}\mathbf{S}$ from Theorem 2.11, we obtain the following corollary.

Corollary 5.37. *For every variety of finite semigroups* $\mathbf{V} \not\subseteq \mathbb{L}\mathbb{D}\mathbf{S}$*, the decision problem* CAYLEY–INTERSECTION$_1(\mathbf{V})$ *is* PSPACE*-complete. For every variety of finite monoids* $\mathbf{V} \not\subseteq \mathbb{D}\mathbf{S}$*, the decision problem* CAYLEY–INTERSECTION$_1(\mathbf{V})$ *is* PSPACE*-complete.*

Unfortunately, it is not known whether there are other varieties of finite monoids with PSPACE-hard intersection non-emptiness. However, we do know that intersection non-emptiness is NP-complete for every non-trivial variety of finite monoids.

We will use reductions from 3-SAT to prove NP-hardness. To simplify notation, let us introduce some definitions. For a set of *variables* $X = \{x_1, \ldots, x_k\}$, we let $\overline{X} = \{\overline{x} \mid x \in X\}$ where each $\overline{x}$ is a new symbol. The set of *literals* over X is $X \cup \overline{X}$. A set of literals is a *clause*. An *assignment* $\mathcal{A}\colon X \to \{0,1\}$ of truth values to the variables X can be extended to all literals over X by letting $\mathcal{A}(\overline{x}) = 1 - \mathcal{A}(x)$ and to clauses $C \subseteq X \cup \overline{X}$ by letting $\mathcal{A}(C) = \max\{\mathcal{A}(\ell) \mid \ell \in C\}$. An assignment $\mathcal{A}$ *satisfies* a set of clauses $\{C_1, \ldots, C_n\}$ if $\mathcal{A}(C_j) = 1$ for all $j \in \{1, \ldots, n\}$. For a given assignment $\mathcal{A}\colon X \to \{0,1\}$, we call $w_{\mathcal{A}} = \ell_1 \cdots \ell_k$, where $\ell_i = x_i$ if $\mathcal{A}(x_i) = 1$ and $\ell_i = \overline{x_i}$ otherwise, the *word induced by* $\mathcal{A}$.

Theorem 5.38. *Let* M *be a non-trivial finite monoid. Then, the decision problem* CAYLEY–INTERSECTION$_7(\{M \times M \times M\})$ *is* NP*-hard.*

Proof. The hardness result is established by providing a polynomial-time reduction of 3-SAT to CAYLEY–INTERSECTION$_7(\{M \times M \times M\})$. We fix some arbitrary element $m \in M \setminus \{1\}$. Suppose we are given a set of variables $X = \{x_1, \ldots, x_k\}$ and a set of clauses $\{C_1, \ldots, C_n\}$ where $C_j = \{\ell_{j1}, \ell_{j2}, \ell_{j3}\}$ for each $j \in \{1, \ldots, n\}$ and for literals $\ell_{j1}, \ell_{j2}, \ell_{j3}$ over X.

We introduce morphisms $g_1, \ldots, g_k, h_1, \ldots, h_n\colon (X \cup \overline{X})^+ \to M \times M \times M$ defined by

$$g_i(\ell) = \begin{cases} (m,1,1) & \text{if } \ell = x_i, \\ (1,m,1) & \text{if } \ell = \overline{x_i}, \\ (1,1,1) & \text{otherwise,} \end{cases} \qquad h_j(\ell) = \begin{cases} (m,1,1) & \text{if } \ell = \ell_{j1}, \\ (1,m,1) & \text{if } \ell = \ell_{j2}, \\ (1,1,m) & \text{if } \ell = \ell_{j3}, \\ (1,1,1) & \text{otherwise} \end{cases}$$

for $\ell \in X \cup \overline{X}$ and for $0 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant n$. We define accepting sets $P = \{(m,1,1),(1,m,1)\}$ and $Q = \{1,m\} \times \{1,m\} \times \{1,m\} \setminus \{(1,1,1)\}$. The claim is that the intersection

$$L = \bigcap_{i=1}^{k} g_i^{-1}(P) \ \cap \ \bigcap_{j=1}^{n} h_j^{-1}(Q)$$

is non-empty if and only if there exists a satisfying assignment for the given variables and clauses.

It is easy to check that if $\mathcal{A}$ is a satisfying assignment, then the word induced by $\mathcal{A}$ indeed belongs to L. Conversely, suppose that the intersection is non-empty and let $u \in L$. We define an assignment $\mathcal{A}\colon X \to \{0,1\}$ by setting $\mathcal{A}(x_i) = 1$ if and only if $g_i(u) = (m,1,1)$ for $1 \leqslant i \leqslant k$. Note that since $u \in L$, this also means that $g_i(u) = (1,m,1)$ whenever $\mathcal{A}(x_i) = 0$. Consider some arbitrary clause C_j and suppose that the first component of $h_j(u)$ is the element m. If $\ell_{j1} = x_i$ for some $i \in \{1,\dots,k\}$, then $g_i(u) = (m,1,1)$, thus $\mathcal{A}(x_i) = 1$. Similarly, if $\ell_{j1} = \overline{x_i}$ for some $i \in \{1,\dots,k\}$, then $g_i(u) = (1,m,1)$, thus $\mathcal{A}(x_i) = 0$. The cases that the second or third components of $h_j(u)$ are m are analogous. Thus, $\mathcal{A}$ is a satisfying assignment, as desired. □

As a next step, we will show that even for varieties of finite semigroups not containing any non-trivial monoids, the problem is NP-hard as soon as the variety contains cyclic semigroups of arbitrarily large cardinality. Recall that such varieties are said to have *unbounded index.*

Let X be a set of variables and let $Y = X \cup \overline{X}$ be the set of literals over X. For a word $w \in Y^+$, the *assignment induced by* w is the mapping $\mathcal{A}_w\colon X \to \{0,1\}$ defined by $\mathcal{A}_w(x) = 1$ if and only if $w \in Y^*xY^*$ for $x \in X$. Assuming that the word w satisfies $\{w\} \cap Y^*x_iY^* \cap Y^*\overline{x_i}Y^* = \emptyset$ for all $i \in \{1,\dots,k\}$, this assignment also satisfies $\mathcal{A}_w(x) = 0$ whenever $w \in Y^*\overline{x}Y^*$.

Theorem 5.39. *Let* $\mathbf{V}$ *be a variety of finite semigroups of unbounded index. Then, the decision problem* Cayley–Intersection($\mathbf{V}$) *is* NP*-hard.*

Proof. We may assume $\mathbf{V} \subseteq \mathbb{L}\mathbf{I}$, otherwise Cayley–Intersection($\mathbf{V}$) is NP-hard by Theorem 5.38 and Theorem 2.11. For each $i \in \mathbb{N}$ the semigroup $C_{i,1}$ belongs to $\mathbf{V}$. To see this, take some arbitrary $i \in \mathbb{N}$. Since $\mathbf{V}$ has unbounded index, some cyclic semigroup T of cardinality at least i appears as a subsemigroup in $\mathbf{V}$. Let s be a generator of T and let a be a generator of $C_{i,1}$. Since $\mathbf{V} \subseteq \mathbb{L}\mathbf{I}$, the period of s is 1. Therefore, the mapping $h\colon T \to C_{i,1}$ defined by $h(s) = a$ is a surjective morphism. By closure of $\mathbf{V}$ under divisors, the semigroup $C_{i,1}$ itself belongs to $\mathbf{V}$.

We now reduce 3-SAT to Cayley–Intersection($\mathbf{V}$). Suppose we are given a set of variables $X = \{x_1,\dots,x_k\}$ and a set of clauses $\{C_1,\dots,C_n\}$ where $C_j = \{\ell_{j1},\ell_{j2},\ell_{j3}\}$ for each $j \in \{1,\dots,n\}$ and for literals $\ell_{j1},\ell_{j2},\ell_{j3}$ over X.

Let S be the cyclic semigroup of cardinality $k+2$ and period 1. To simplify notation, we will identify the elements of S with the set $\{1,\dots,k+2\}$ and write the operation of S additively. The generator of S is denoted by 1 and the operation is then given by $i + j = \min\{i + j, k + 2\}$ for $i, j \in \{1, \dots, k + 2\}$. We introduce morphisms $g_0, \dots, g_k, h_1, \dots, h_n \colon (X \cup \overline{X})^+ \to S$ defined by

$$g_i(\ell) = \begin{cases} 2 & \text{if } i > 0 \text{ and } \ell \in \{x_i, \overline{x_i}\}, \\ 1 & \text{otherwise,} \end{cases} \qquad h_j(\ell) = \begin{cases} 3 & \text{if } \ell \in C_j, \\ 1 & \text{otherwise} \end{cases}$$

for $\ell \in X \cup \overline{X}$, for $0 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant n$. We let $P_0 = \{k\}$, $P_1 = \cdots = P_k = \{k + 1\}$ and $Q = \{k + 2\}$. It is easy to check that the intersection

$$L = \bigcap_{i=0}^{k} g_i^{-1}(P_i) \ \cap \ \bigcap_{j=1}^{n} h_j^{-1}(Q)$$

is non-empty if and only if there is a satisfying assignment. We observe that

1. $g_0^{-1}(P_0)$ contains all words over $X \cup \overline{X}$ with exactly k letters,
2. $g_i^{-1}(P_i) \cap g_0^{-1}(P_0)$ contains all words from the set $(X \cup \overline{X})^k$ with exactly one occurrence of x_i or exactly one occurrence of $\overline{x_i}$ (but not both), and
3. $h_j^{-1}(Q_j) \cap g_0^{-1}(P_0)$ contains all words from the set $(X \cup \overline{X})^k$ with at least one occurrence of any of the literals $\ell_{j1}, \ell_{j2}, \ell_{j3}$.

By the first two properties, all words from L are of the form $\ell_1 \cdots \ell_k \in (X \cup \overline{X})^k$ with $|\{\ell_1, \dots, \ell_k\} \cap \{x_i, \overline{x_i}\}| = 1$ for all $i \in \{1, \dots, k\}$. Thus, the assignment $\mathcal{A}_w$ induced by w assigns 1 to a literal ℓ if and only if ℓ occurs in w. Now, if $w \in L$, by the third property, we have $\mathcal{A}_w(\ell_{j1}) = 1$ or $\mathcal{A}_w(\ell_{j2}) = 1$ or $\mathcal{A}_w(\ell_{j3}) = 1$ for each $j \in \{1, \dots, n\}$. This shows that $\mathcal{A}_w$ is satisfying. Conversely, if there exists a satisfying assignment $\mathcal{A} \colon X \to \{0, 1\}$, the word induced by $\mathcal{A}$ is contained in L.

It is obvious that the reduction can be performed by a log-space transducer. □

A special case of the previous theorem is that intersection non-emptiness for the variety $\bigcup_k \mathbb{L}\mathbf{I}_k = \mathbb{L}\mathbf{I}$ is NP-complete. Since we know by Corollary 5.30 that, for each variety $\mathbb{L}\mathbf{I}_k$, intersection non-emptiness is in $\Sigma_2\mathsf{TIME}(\log n)$, this yields that there is no maximal class of finite semigroups with a tractable intersection non-emptiness problem.

Corollary 5.40. *For every class of finite semigroups* $\mathbf{C}$*, such that the decision problem* Cayley–Intersection($\mathbf{C}$) *is not* NP*-hard, there exists a variety of finite semigroups* $\mathbf{V} \not\subseteq \mathbf{C}$*, such that* Cayley–Intersection($\mathbf{V}$) *belongs to* $\Sigma_2\mathsf{TIME}(\log n) \subseteq \mathsf{AC}^0$.

We conclude the hardness part by investigating another very restricted class of finite semigroups of bounded index (which also played an important role in Section 5.1.2).

Theorem 5.41. *Let* $\mathbf{V}$ *be the variety of finite semigroups defined by the equations* $x^2 = xyx = 0$*. Then,* Cayley–Intersection($\mathbf{V}$) *is* NP*-complete.*

Proof. As in the previous proof, we reduce 3-SAT to Cayley–Intersection($\mathbf{V}$). Containment in NP follows from Corollary 5.33 and from $\mathbf{V} \subseteq \mathbb{L}\mathbf{I}$.

Suppose we are given a set of variables $X = \{x_1, \ldots, x_k\}$ and a set of clauses $\{C_1, \ldots, C_n\}$ where $C_j = \{\ell_{j1}, \ell_{j2}, \ell_{j3}\}$ for each $j \in \{1, \ldots, n\}$ and literals $\ell_{j1}, \ell_{j2}, \ell_{j3}$ over X. Let S be the finite semigroup $\{(i,j) \mid 1 \leqslant i < j \leqslant k+1\} \cup \{0\}$ defined by the multiplication

$$(i,j)(k,\ell) = \begin{cases} (i,\ell) & \text{if } k = j, \\ 0 & \text{otherwise.} \end{cases}$$

The element 0 is the zero element. Let $g, h_1, \ldots, h_n \colon (X \cup \overline{X})^+ \to S$ be the morphisms defined by $g(x_i) = g(\overline{x_i}) = (i, i+1)$ and by

$$h_j(x_i) = \begin{cases} (i,i+1) & \text{if } x_i \notin C_j, \\ 0 & \text{otherwise,} \end{cases} \qquad h_j(\overline{x_i}) = \begin{cases} (i,i+1) & \text{if } \overline{x_i} \notin C_j, \\ 0 & \text{otherwise} \end{cases}$$

for $1 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant n$. As accepting sets, we choose $P = \{(1, k+1)\}$ for g and $Q_1 = \cdots = Q_n = \{0\}$ for $h_1, \ldots, h_n$. Again, we would like to show that the intersection

$$L = g^{-1}(P) \;\cap\; \bigcap_{j=1}^{n} h_j^{-1}(Q_j)$$

is non-empty if and only if there exists a satisfying assignment for $\{C_1, \ldots, C_n\}$. The following two properties hold:

1. $g^{-1}(P)$ contains all words of the form $\ell_1 \cdots \ell_k$ with $\ell_i \in \{x_i, \overline{x_i}\}$ for $1 \leqslant i \leqslant k$,
2. $g^{-1}(P) \cap h_j^{-1}(Q_j)$ contains all words of this form containing at least one of the letters $\ell_{j1}, \ell_{j2}, \ell_{j3}$.

Let $w \in A^+$ be a word with $g(w) \in P$ and $h_j(w) \in Q_j$ for all $j \in \{1, \ldots, n\}$. Then, by the first property above, the assignment $\mathcal{A}_w$ induced by w assigns 1 to a literal ℓ if and only if ℓ occurs in w. Moreover, by the second property, we have $\mathcal{A}_w(C_1) = \cdots = \mathcal{A}_w(C_n) = 1$ and thus, $\mathcal{A}_w$ satisfies $\{C_1, \ldots, C_n\}$. Conversely, it is easy to see that each word induced by a satisfying assignment is contained in L.

Note that the constructed semigroup satisfies $x^2 = xyx = 0$ since by definition, we have $(i,j)(i,j) = (i,j)(k,\ell)(i,j) = 0$ for all $(i,j), (k,\ell) \in S$. It is obvious that the reduction can be performed by a log-space transducer. □

It is interesting to observe that NP-hardness of intersection non-emptiness seems to be caused by a combination of structural and size-related properties: the problem is NP-hard for every variety containing non-trivial monoids, for every (commutative) variety of *unbounded index*, as well as for the variety defined by $x^2 = xyx = 0$ which is non-commutative but has index at most 2. However, for any monoid-free commutative variety with bounded index, the problem is in qAC^0 by Corollary 5.31, and thus not hard for any class containing Parity.

5.5 Variety Membership

The membership problem for a class of finite semigroups $\mathbf{C}$ asks, given a finite semigroup S in Cayley encoding, whether S belongs to $\mathbf{C}$. We are mainly interested in membership to varieties of finite semigroups. It was shown that many natural operations on varieties do not preserve decidability of the membership problem [ABR92, Rho99, Aui10]. Later, people started investigating the computational complexity of the problem [Alm91, Vol95, Alm94, KS95, Vol97, ABKK15, Ver]. While it has been well-known that for many decidable varieties, membership can be tested in polynomial time, it was conjectured that not all varieties have this property. Indeed, in 2006, Jackson and McKenzie constructed a variety for which membership cannot be decided in polynomial time unless $\mathsf{P} = \mathsf{NP}$ [JM06]. Similar constructions followed; see e.g. [JV10].

One of the main tools to show that the membership problem of a given variety $\mathbf{V}$ can be decided in polynomial time is to show that $\mathbf{V}$ can be defined by a finite set of so-called *ω-identities*. The naïve algorithm for testing membership to a variety defined by a finite set of such identities is simply taking all possible assignments of elements to the variables in the identities, computing the left-hand and right-hand sides and testing whether all equalities hold. Each left-hand and right-hand side can actually be computed within logarithmic space. Thus, definability by a finite set of *ω-identities* also yields decidability in deterministic log-space [SW15, Theorem 2.19]. Theorem 3.2 and Proposition 3.6 actually yield the following two stronger statements.

Corollary 5.42. *Let $\mathbf{C}$ be a $\Sigma_k[\,\cdot\,]$-definable class of finite semigroups. Then the membership problem for $\mathbf{C}$ is in $\mathsf{\Sigma_k TIME}(\log n) \subseteq \mathsf{AC}^0$.*

Corollary 5.43. *If $\mathbf{V}$ is definable by a finite set of ω-identities, then the membership problem for $\mathbf{V}$ is in $\mathsf{\Pi_2 TIME}(\log n) \subseteq \mathsf{AC}^0$.*

By the results listed in Section 3.1, we also obtain efficient decidability for varieties such as the variety of finite solvable groups. Thus, there are examples of varieties whose membership problems are undecidable, NP-complete or in the logarithmic time hierarchy. It is natural to ask whether there are classes of semigroups with membership complete for other common complexity classes. The following proposition answers this positively.

Proposition 5.44. *For every language L, there exists a class of finite semigroups $\mathbf{C}$ such that L is reducible to the membership problem for $\mathbf{C}$ via $\mathsf{DLOGTIME}$-uniform AC^0 reductions, and equivalent to the membership problem for $\mathbf{C}$ via log-space reductions.*

Proof. Since every language is reducible to a non-trivial language over a binary alphabet via $\mathsf{DLOGTIME}$-uniform AC^0 reductions, we may assume that $L \subsetneq A^+$ with $A = \{a, b\}$.

To each word $w = a_1 \cdots a_\ell$ (with $a_1, \dots, a_\ell \in A$), we assign a finite semigroup $S_w = \{(i, 0) \mid 1 \leqslant i \leqslant \ell\} \cup \{(i, 1) \mid 1 \leqslant i \leqslant \ell \text{ and } a_i = b\}$ with the multiplication

$$(i,j)(k,\ell) = \begin{cases} (i,j) & \text{if } k < i, \\ (i, j+\ell \bmod 2) & \text{if } k = i, \\ (k,\ell) & \text{if } k > i. \end{cases}$$

Let $\mathbf{C} = \{S_w \mid w \in L\}$. Clearly, on input w, we can construct each entry of the multiplication table of S_w by a deterministic random-access Turing machine in logarithmic time: we can compute the maximum of two $\mathcal{O}(\log n)$ bit numbers, add single bits modulo 2 and access input letters by address. This yields a $\mathsf{DLOGTIME}$-uniform AC^0 reduction from L to the membership problem for $\mathbf{C}$.

It remains to prove that the membership problem for $\mathbf{C}$ is reducible to L via log-space reductions. Suppose we are given a finite semigroup S. We first verify that the semigroup is commutative. Then, we check whether for all $s, t \in S$ with $s \neq t$, we have $s^3 = s$ and $st \in \{s, t\}$. By Corollary 5.42, each of these tests can be performed by an alternating random-access Turing machine in logarithmic time. If any of the tests are not passed, we reject the input, i.e., output some fixed word not in L. Otherwise, we can order the idempotents of S linearly by $e < f$ whenever $ef = f$. We successively enumerate the idempotents of S in increasing order and check for every idempotent e, whether the $\mathcal{H}$-class of e has size 1 or 2. If the size is 1, we output the letter a, otherwise we output b. Note that this enumeration can be done in deterministic log-space.

It is easy to verify that the machine rejects (i.e., outputs the fixed word not in L) if and only if S is not isomorphic to any semigroup in $\{S_w \mid w \in A^+\}$, and it produces the output word w if and only if S is isomorphic to S_w. □

Note that the classes of semigroups constructed in the proposition are quite artificial and not even closed under taking subsemigroups in general. This leads to the question of whether one can also find *varieties* with membership problems complete for other classes which turns out to be a much more challenging task. While we do not have a general result as in the case of arbitrary classes of finite semigroups, we will demonstrate that the membership problem for the variety $\mathbb{E}\mathbf{A}$ is L-complete.

We make some preliminary considerations. Firstly, it is well-known that each regular $\mathcal{J}$-class of a finite semigroup S forms a partial semigroup which is isomorphic to a Rees matrix semigroup; see e.g. [RS09, Section A.4]. Moreover, a finite semigroup S belongs to $\mathbb{E}\mathbf{A}$ if and only if, for each regular $\mathcal{J}$-class J of S, the Rees matrix semigroup corresponding to J belongs to $\mathbb{E}\mathbf{A}$ [RS09, Corollary 4.13.4]. This allows us to confine ourselves to Rees matrix semigroups.

The *incidence graph* of a Rees matrix semigroup with structure group G, index sets A and B, and sandwich matrix $C\colon B \times A \to G \cup \{0\}$ is defined as the edge-labelled graph with vertex set $A \cup B$, edge set $E = \{(b, a) \in B \times A \mid C(b, a) \neq 0\}$ and the labels as given by C. For Rees matrix semigroups without zero, the definition is the same and we have $E = B \times A$. In any case, each edge label in the incidence graph is an element from G. We consider undirected simple cycles in this graph, i.e., sequences $(c_1, \dots, c_k)$ where $c_1, \dots, c_k \in A \cup B$ are pairwise disjoint vertices with $(c_i, c_{i+1}) \in E$ or $(c_{i+1}, c_i) \in E$ for all $i \in \{1, \dots, k-1\}$, and where $c_k = c_1$. The edge labelling can be extended to $E \cup \{(a, b) \mid (b, a) \in E\}$ by setting $C(a, b) = (C(b, a))^{-1}$. The label of an undirected simple cycle $(c_1, \dots, c_k)$ is the product $C(c_1, c_2)C(c_2, c_3) \cdots C(c_{k-1}, c_k)$ in G. The subsequent construction will be based on the following consequence of Graham's Theorem; see e.g. [RS09, Corollary 4.13.24] for a proof.

Theorem 5.45. *A Rees matrix semigroup belongs to* $\mathbb{E}\mathbf{A}$ *if and only if each undirected simple cycle in its incidence graph is labelled by the identity element* 1.

The proof of NL-hardness of the membership problem for $\mathbb{E}\mathbf{A}$ uses a reduction from connectivity in undirected graphs.

Theorem 5.46. *The membership problem for the varieties* $\mathbb{E}\mathbf{A}$ *and* $\mathbf{A} \vee \mathbf{G}$ *is* L*-hard under* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions.*

Proof. We reduce the complement of the reachability problem in undirected graphs to membership in $\mathbb{E}\mathbf{A}$, i.e., we describe how to convert an undirected graph $G = (V, E)$ and vertices $s, t \in V$ into a Rees matrix semigroup $S = \mathcal{M}^0(C_{1,2}, V, V, C)$ such that $S \in \mathbb{E}\mathbf{A}$ if and only if t is not reachable from s in G. Without loss of generality, we may assume that $s \neq t$ and $\{s,t\} \notin E$.

The structure group of S is the cyclic group $C_{1,2} = \{1, a\}$ with $a^2 = 1$. Both index sets are V. The sandwich matrix $C \colon V \times V \to C_{1,2} \cup \{0\}$ is given by

$$C(v,w) = \begin{cases} 1 & \text{if } v = w \text{ or } \{v,w\} \in E, \\ a & \text{if } \{v,w\} = \{s,t\}, \\ 0 & \text{otherwise.} \end{cases}$$

The correctness of the construction follows immediately from Theorem 5.45. It is clear that every entry of the Cayley table can be computed by a deterministic logarithmic-time random-access Turing machine. Thus, the reduction can be performed by a $\mathsf{DLOGTIME}$-uniform family of AC^0 circuits.

A Rees matrix semigroup belongs to $\mathbb{E}\mathbf{A}$ if and only if it belongs to $\mathbf{A} \vee \mathbf{G}$; see e.g. [RS09, Theorem 4.13.31]. □

Since we know that L-hard problems cannot be decided in AC^0, we obtain the following corollary.

Corollary 5.47. *Neither* $\mathbb{E}\mathbf{A}$ *nor* $\mathbf{A} \vee \mathbf{G}$ *are* $\mathrm{FO}[\,\cdot\,]$*-definable. In particular, neither of these varieties can be defined by a finite set of* ω*-identities.*

We remark that the previous two results also hold for the semidirect product $\mathbf{A} * \mathbf{G}$ (which was not introduced formally). Since both $\mathbf{A}$ and $\mathbf{G}$ are definable by a finite set of ω-identities, this yields a new proof that the $\mathbb{E}$-operator, joins and semidirect products do not preserve $\mathrm{FO}[\,\cdot\,]$-definability. Our proof for containment in L heavily relies on the incidence graph representation of Rees matrix semigroups and also requires Reingold's deterministic log-space algorithm for reachability in undirected graphs [Rei08].

Theorem 5.48. *The membership problem for the variety* $\mathbb{E}\mathbf{A}$ *is* L*-complete under* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reductions.*

Proof. In view of Theorem 5.46, it remains to show that the membership problem for $\mathbb{E}\mathbf{A}$ belongs to L. As explained above, a finite semigroup S belongs to $\mathbb{E}\mathbf{A}$ if and only if, for each regular $\mathcal{J}$-class J of S, the Rees matrix semigroup corresponding to J

belongs to $\mathbb{E}\mathbf{A}$. The construction of this Rees matrix semigroup representation for a given regular $\mathcal{J}$-class can be performed by a log-space transducer. This observation allows us to reduce the problem to the case of Rees matrix semigroups: we iterate through all regular $\mathcal{J}$-classes of our input semigroup and can think of having their Rees matrix semigroup representations available.

We will leverage Theorem 5.45 and work on the incidence graph of each Rees matrix semigroup. Again, it is easy to see that the incidence graph (V, E) can be computed by a log-space transducer from the Rees matrix semigroup representation. The edge labels will be denoted by C.

The key observation is that in order to certify or refute the existence of an undirected simple cycle labelled by 1, we do not need to consider all simple cycles: it suffices to choose any spanning forest $F \subseteq E$ and then compute, for every edge $(b, a) \in E \setminus F$, the label of the simple cycle obtained by starting with (b, a) and then walking along spanning tree edges (and reverse spanning tree edges) from a back to b. The edges in $E \setminus F$ are called *bridges*. For a bridge $(b, a) \in E \setminus F$, the simple undirected cycle *induced by* (b, a) is the unique simple cycle consisting of the edge (b, a) and spanning tree edges (and reverse spanning tree edges) from a to b. If every induced cycle has label 1, we can repeatedly replace bridges in *any* simple undirected cycle by spanning tree edges without changing its label (and remove redundant subpaths of the form (a, b, a) or (b, a, b) with $a \in A$, $b \in B$), until the cycle becomes an induced cycle itself.

We choose the lexicographically first spanning forest F of the incidence graph: checking whether an edge (b', a') belongs to F amounts to testing whether b' is reachable from a' in the (undirected) graph restricted to edges smaller than (b', a') — the edge-ordering is given implicitly by the way the graph is produced by the log-space transducer. For each edge $(b, a) \in E \setminus F$, we check whether the product of the labels of (b, a) and the corresponding (reverse) spanning tree edges leading from a to b equals 1 in G. This is done using the standard deterministic log-space algorithm for tree traversal [CM87], multiplying the labels while walking along the edges.

More explicitly, to enumerate all bridges, we iterate over all tuples $(b, a) \in B \times A$ and ask whether or not (b, a) belongs to the spanning forest as described above. The process of walking along the simple undirected cycle induced by a bridge (b, a) works as follows: when starting at the vertex a, we look at all incident edges and take the lexicographically smallest among those which are part of the spanning forest F. We walk along this edge. Whenever we arrive at a vertex c, we again look at all incident edges and choose the lexicographically smallest edge which belongs to F and is larger than the edge we just used to enter c. If no such edge exists, we choose the lexicographically smallest edge of F incident to c instead. Note that if we arrive at a leaf, we immediately go back to the vertex from which the leaf was entered. The process can be thought of as tracing an Euler tour *around* the spanning tree. We eventually arrive at b because it belongs to the same connected component as a. Whenever walking along an edge (b', a'), we multiply the currently stored label of the path by the label $C(b', a')$ of the edge. When walking along such an edge in the opposite direction, we multiply the currently stored label by $C(a', b')$. Since $C(b', a')$ and $C(a', b')$ are mutually inverse, multiplications are inverted in the corresponding backtracking steps. □

As a final remark, it is important to note that our techniques cannot be used to decide whether a recognizable language given by a morphism belongs to a certain *variety of languages*, i.e., a class of languages of the form $\mathbf{V}(A^+)$ for some alphabet A and some variety of finite semigroups $\mathbf{V}$. This problem is NL-hard for every non-trivial variety.

Proposition 5.49. *Let $\mathbf{V} \subsetneq \mathbf{S}$ be a non-trivial variety of finite semigroups. Then, the problem of deciding whether the language represented by a recognizing morphism to a finite semigroup in Cayley encoding belongs to $\mathbf{V}(A^+)$ is NL-hard under $\mathsf{DLOGTIME}$-uniform AC^0 reductions.*

Proof. We prove NL-hardness by a reduction from the complement of s-t-connectivity in directed graphs. Let $G = (V, E)$ be a directed graph and let $s, t \in V$ with $s \neq t$.

Let $L \subseteq B^+$ be a recognizable language which is not recognized by any semigroup from $\mathbf{V}$, let $g \colon B^+ \to T$ be a morphism to a finite semigroup $T \notin \mathbf{V}$ and let $Q \subseteq T$ such that $g^{-1}(Q) = L$. Such a language and such a morphism exist by Eilenberg's Theorem on $+$-*varieties of languages* [Eil76, Theorem 3.4s] and the assumption that $\mathbf{V}$ is not the class of all finite semigroups. We extend g to a monoid morphism $g \colon B^* \to T^1$. Below, the symbol $\circ$ will be used to denote the multiplication in T^1.

We define a semigroup $S = V \times T^1 \times V \cup \{0\}$ by the binary operation

$$(v, m, w) \cdot (x, n, y) = \begin{cases} (v, m \circ n, y) & \text{if } w = x, \\ 0 & \text{otherwise.} \end{cases}$$

The element 0 is a zero. Let $A = \{(v, \varepsilon, w) \mid (v, w) \in E \text{ and } v \neq t\} \cup \{(t, b, t) \mid b \in B\}$. Let $h \colon A^+ \to S$ be the morphism defined by $h(v, b, w) = (v, g(b), w)$ for $(v, b, w) \in A$ and let $P = \{(s, q, t) \mid q \in Q\}$.

Clearly, every word in $h^{-1}(P)$ yields a path from s to t in G. The empty set is recognized by every finite semigroup. In particular, if t is not reachable from s, then $h^{-1}(P) = \emptyset$ is recognized by a semigroup from $\mathbf{V}$.

Now, suppose that there exists a directed path $(x_1, \dots, x_k)$ with $x_1 = s$ and $x_k = t$. We may assume that $t \notin \{x_1, \dots, x_{k-1}\}$. Assume, for the sake of contradiction, that the language $h^{-1}(P)$ is recognized by a finite semigroup $T' \in \mathbf{V}$, i.e., there exist a morphism $f \colon A^+ \to T'$ and a set $R \subseteq T'$ with $f^{-1}(R) = h^{-1}(P)$. We let $g' \colon B^+ \to T'$ be the morphism defined by $g'(b) = f(t, b, t)$ for all $b \in B$ and we let $Q' = \big\{ r \in T' \bigm| f\big((x_1, \varepsilon, x_2) \cdots (x_{k-1}, \varepsilon, x_k)\big) r \in R \big\}$. We obtain

$$\begin{aligned} g(b_1 \cdots b_k) \in Q \quad &\Leftrightarrow \quad h\big((x_1, \varepsilon, x_2) \cdots (x_{k-1}, \varepsilon, x_k)(t, b_1, t) \cdots (t, b_k, t)\big) \in P \\ &\Leftrightarrow \quad f\big((x_1, \varepsilon, x_2) \cdots (x_{k-1}, \varepsilon, x_k)(t, b_1, t) \cdots (t, b_k, t)\big) \in R \\ &\Leftrightarrow \quad g'(b_1 \cdots b_k) \in Q' \end{aligned}$$

for all sequences of elements $b_1, \dots, b_k \in B$. Therefore, the preimage of Q' under the morphism g' is L, contradicting the choice of L.

Since the semigroup T is fixed, it is clear that every entry of the Cayley table of S, every image of a letter under h and every entry of P can be computed in $\mathsf{DLOGTIME}$. We assume that the graph is given in a suitable encoding, such as an adjacency matrix with entries whose addresses are $\mathsf{DLOGTIME}$-computable. □

Chapter 6

Conclusion and Future Work

We investigated decision problems on semigroups and recognizable languages. We mainly considered variants where the languages in the input are given as morphisms to finite semigroups and the semigroups are given as multiplication tables. We showed that emptiness, universality, inclusion, equivalence and finiteness all reduce to the Cayley semigroup membership problem which is NL-complete in the general case. We showed that the problem remains NL-complete when restricting the input to any non-completely regular, non-commutative variety of finite semigroups. We also proved that for a variety of finite monoids $\mathbf{V}$, the problem is in $\mathsf{NPOLYLOGTIME}$ if and only if $\mathbf{V}$ either only contains Clifford semigroups or only contains commutative semigroups, thereby resolving a generalization of a long-standing open problem in complexity theory. We showed how to solve the case of regular bands in $\Pi_2\mathsf{TIME}(\log n)$, a low level of the logarithmic-time hierarchy. This leaves only a small class of non-completely regular, non-commutative varieties for further investigation. As a first step to tackling this problem, we suggest considering the variety of finite bands. It would also be interesting to see whether the complexity results for Clifford semigroups and for commutative semigroups can be extended to other complexity classes. Is Cayley semigroup membership for commutative semigroups decidable in deterministic log-space? Is the problem for Clifford semigroups decidable in FOLL? For varieties of finite semigroups, we showed that a dichotomy result, as obtained for monoids, is impossible: if a class of semigroups contains all varieties with Cayley semigroup membership in $\mathsf{NLOGTIME}$, it already contains a variety whose Cayley semigroup membership problem is NL-complete.

We also considered the intersection non-emptiness problem for languages recognized by finite semigroups. We showed that the problem is NP-hard for any non-trivial variety of finite monoids. As in the Cayley semigroup membership setting, for semigroups, there is no classification in terms of a maximal class for which the problem is tractable. We gave examples of varieties for which the intersection non-emptiness problem is in AC^0 or in qAC^0. We showed that PSPACE-completeness holds for every variety not contained in $\mathbb{LDS}$ and gave NP algorithms for subclasses of $\mathbb{LDS}$. For varieties of finite monoids $\mathbf{V}$, the problem is PSPACE-complete whenever $\mathbf{V} \not\subseteq \mathbb{DS}$. With the current techniques, a dichotomy result seems out of reach: resolving the question of whether or not the varieties $\mathbb{DS}$ and $\mathbb{LDS}$ have the poly PCP is likely to be closely linked to the problem of finding a suitable language characterization for $\mathbb{DS}$ which is considered difficult and has been open for at least 25 years [Alm94]. As a first step, we suggest investigating the complexity of Cayley–Intersection($\mathbb{L}\mathbf{G}$). Does $\mathbb{L}\mathbf{G}$ admit the poly PCP? Theorem 5.6 implies that $\mathbb{L}\mathbf{G}$ does not admit the polylog CP, so we cannot apply the transfer lemma as in the case of groups.

For variety membership, we introduced the notion of FO$[\,\cdot\,]$-definability of a class of semigroups. This concept is interesting on its own and we hope to trigger further research in that area. Are there other non-trivial examples of FO$[\,\cdot\,]$-definable varieties which are not captured by the closure properties and methods presented in this work? Does quantifier alternation capture a strict hierarchy of varieties of finite semigroups, i.e., is there, for every $k \in \mathbb{N}$, a variety which is $\Sigma_{k+1}[\,\cdot\,]$-definable but not $\Sigma_k[\,\cdot\,]$-definable? Can we capture more classes if we consider an enriched logic with additional, more sophisticated, yet natural predicates? Are there varieties of finite semigroups whose membership problem is complete for other complexity classes such as NL or P?

We also note that there is a lot of closely related work on complexity questions for finite semigroups. For example, the *separation* and *covering* problems attracted quite some attention in recent years since they turned out to be not only a natural extension of the membership problem but also helped with obtaining new decidability and complexity results for some levels of the *dot-depth hierarchy.* As shown in [PZ18], the complexity of separation is independent of whether the languages are represented as DFAs or as monoids in Cayley encoding. An interesting open problem in this area is the complexity of separation for aperiodic semigroups. To date, this problem is only known to be PSPACE-hard and decidable in exponential time.

Acknowledgments

There are many people who contributed to the successful completion of this thesis in one way or another. First and foremost, I would like to thank Manfred Kufleitner for introducing me to the field of finite semigroup theory, for his suggestion to work on algorithms for recognizing morphisms, and for numerous helpful discussions in the early stages of my research. I am very grateful to Volker Diekert for offering me a position at the Institute for Formal Methods in Computer Science (FMI) of the University of Stuttgart, and for his constant support, advice and encouragement. I would also like to thank Pascal Weil who kindly agreed to co-examine this thesis.

I want to express my gratitude to all former and current members of the FMI for a great time and for many inspiring conversations. Special thanks to Carlos Camino, Jan Philipp Wächter and Armin Weiß for proofreading and reviewing earlier conference versions of some of the results presented in this thesis. In addition, I am deeply indebted to everybody who kindly agreed to read parts of this thesis and suggested grammatical and stylistic improvements: Cassandra Duguay, Uwe Fleischer, Julia Jäger and Tobias Rupp — thank you so much! Unfortunately, there is not enough space to list all valuable comments I received from people outside the University of Stuttgart. Each of them is highly appreciated! I would specifically like to mention Charles Paperman who shared an idea which lead to the proof of Lemma 3.5.

Last but not least, I would like to thank my family, my friends and Julia for their unwavering patience and support throughout the years!

Bibliography

[ABKK15] Jorge Almeida, Jana Bartoňová, Ondřej Klíma, and Michal Kunc. On decidability of intermediate levels of concatenation hierarchies. In *DLT 2015, Proceedings*, pages 58–70, Cham, 2015. Springer. (Cited on page 87.)

[ABR92] Douglas Albert, Robert Baldinger, and John L. Rhodes. Undecidability of the identity problem for finite semigroups. *The Journal of Symbolic Logic*, 57:179–192, 1992. (Cited on page 87.)

[Alm91] Jorge Almeida. On the membership problem for pseudovarieties of commutative semigroups. *Semigroup Forum*, 42(1):47–51, Dec 1991. (Cited on page 87.)

[Alm94] Jorge Almeida. *Finite Semigroups and Universal Algebra.* World Scientific, Singapore, 1994. (Cited on pages 17, 18, 20, 25, 87, and 93.)

[Aui10] K. Auinger. On the decidability of membership in the global of a monoid pseudovariety. *International Journal of Algebra and Computation*, 20(02):181–188, 2010. (Cited on page 87.)

[AV04] V. Arvind and T. C. Vijayaraghavan. Abelian permutation group problems and logspace counting classes. In *CCC 2004, Proceedings*, pages 204–214, June 2004. (Cited on page 65.)

[Bar89] David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *J. Comput. Syst. Sci.*, 38(1):150–164, 1989. (Cited on pages 10 and 71.)

[Bau65] Gilbert Baumslag. Residual nilpotence and relations in free groups. *Journal of Algebra*, 2(3):271 – 282, 1965. (Cited on page 20.)

[BCST92] David A. Mix Barrington, Kevin J. Compton, Howard Straubing, and Denis Thérien. Regular languages in NC^1. *J. Comput. Syst. Sci.*, 44(3):478–499, 1992. (Cited on page 10.)

[Bea88a] Martin Beaudry. Membership testing in commutative transformation semigroups. *Inf. Comput.*, 79(1):84–93, 1988. (Cited on page 64.)

[Bea88b] Martin Beaudry. *Membership Testing in Transformation Monoids.* PhD thesis, McGill University, Montreal, Quebec, 1988. (Cited on pages 64 and 69.)

[Bea94] Martin Beaudry. Membership testing in threshold one transformation monoids. *Inf. Comput.*, 113(1):1–25, 1994. (Cited on page 64.)

[Ber97] László Bernátsky. Regular expression star-freeness is PSPACE-complete. *Acta Cybernetica*, 13(1):1–21, 1997. (Cited on pages 64 and 78.)

[BIP98] P. Beame, R. Impagliazzo, and T. Pitassi. Improved depth lower bounds for small distance connectivity. *Computational Complexity*, 7(4):325–345, Dec 1998. (Cited on page 32.)

[Bir70] P. A. Birjukov. Varieties of idempotent semigroups. *Algebra i Logika*, 9:255–273, 1970. (Cited on page 21.)

[BIS90] David A. Mix Barrington, Neil Immerman, and Howard Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41(3):274 – 306, 1990. (Cited on pages 30, 32, and 38.)

[BKLM01] David Mix Barrington, Peter Kadau, Klaus-Jörn Lange, and Pierre McKenzie. On the complexity of some problems on groups input as multiplication tables. *Journal of Computer and System Sciences*, 63(2):186–200, 2001. (Cited on pages 11 and 64.)

[BLS87] László Babai, Eugene M. Luks, and Ákos Seress. Permutation groups in NC. In *STOC 1987, Proceedings*, pages 409–420, 1987. (Cited on pages 64 and 65.)

[BM91] David A. Mix Barrington and Pierre McKenzie. Oracle branching programs and Logspace versus P. *Information and Computation*, 95(1):96–115, 1991. (Cited on pages 11, 64, and 65.)

[BMT92] Martin Beaudry, Pierre McKenzie, and Denis Thérien. The membership problem in aperiodic transformation monoids. *J. ACM*, 39(3):599–616, 1992. (Cited on pages 10, 64, and 69.)

[BS84] L. Babai and E. Szemeredi. On the complexity of matrix group problems I. In *FOCS 1984, Proceedings*, pages 229–240, Oct 1984. (Cited on page 52.)

[BT88] David A. Mix Barrington and Denis Thérien. Finite monoids and the fine structure of NC^1. *J. ACM*, 35:941–952, 1988. (Cited on page 10.)

[CH91] Sung Cho and Dung T. Huynh. Finite automaton aperiodicity is PSPACE-complete. *Theoretical Computer Science*, 88:96–116, 1991. (Cited on pages 64 and 78.)

[Chl86] Bogdan S. Chlebus. Domino-tiling games. *Journal of Computer and System Sciences*, 32(3):374–392, 1986. (Cited on page 82.)

[CM87] Stephen A Cook and Pierre McKenzie. Problems complete for deterministic logarithmic space. *Journal of Algorithms*, 8(3):385–394, 1987. (Cited on pages 71 and 90.)

[COST16] Xi Chen, Igor C. Oliveira, Rocco A. Servedio, and Li-Yang Tan. Near-optimal small-depth lower bounds for small distance connectivity. In *STOC 2016, Proceedings*, pages 612–625, 2016. (Cited on page 32.)

[DGH05] Volker Diekert, Claudio Gutiérrez, and Christian Hagenah. The existential theory of equations with rational constraints in free groups is PSPACE-complete. *Information and Computation*, 202:105–140, 2005. (Cited on pages 64 and 78.)

[Eil76] Samuel Eilenberg. *Automata, Languages, and Machines*, volume B. Academic Press, New York and London, 1976. (Cited on page 91.)

[Fen71] C. F. Fennemore. All varieties of bands I, II. *Mathematische Nachrichten*, 48:237–252, 253–262, 1971. (Cited on page 21.)

[FHL80] M. Furst, J. Hopcroft, and E. Luks. Polynomial-time algorithms for permutation groups. In *SFCS 1980, Proceedings*, pages 36–41, Oct 1980. (Cited on page 64.)

[FK15] Lukas Fleischer and Manfred Kufleitner. Efficient Algorithms for Morphisms over Omega-Regular Languages. In *FSTTCS 2015, Proceedings*, volume 45 of *LIPIcs*, pages 112–124. Dagstuhl Publishing, 2015. (Cited on pages 9 and 13.)

[FK16] Lukas Fleischer and Manfred Kufleitner. Operations on Weakly Recognizing Morphisms. In *DCFS 2016, Proceedings*, volume 9777 of *LNCS*, pages 126–137. Springer, 2016. (Cited on pages 13 and 76.)

[FK17] Lukas Fleischer and Manfred Kufleitner. Green's Relations in Finite Transformation Semigroups. In *CSR 2017, Proceedings*, volume 10304 of *LNCS*, pages 112–125. Springer, 2017. (Cited on page 13.)

[FK18a] Lukas Fleischer and Manfred Kufleitner. The complexity of weakly recognizing morphisms. *RAIRO-Theor. Inf. Appl.*, 2018. (Cited on page 13.)

[FK18b] Lukas Fleischer and Manfred Kufleitner. Green's Relations in Deterministic Finite Automata. *Theory of Computing Systems*, 2018. (Cited on pages 13 and 29.)

[FK18c] Lukas Fleischer and Manfred Kufleitner. The Intersection Problem for Finite Monoids. In *STACS 2018, Proceedings*, pages 30:1–30:14. Dagstuhl Publishing, 2018. (Cited on pages 12, 13, 57, 60, 78, 80, and 82.)

[FK18d] Lukas Fleischer and Manfred Kufleitner. Testing Simon's congruence. In *MFCS 2018, Proceedings*, volume 117 of *LIPIcs*, pages 62:1–62:13. Dagstuhl Publishing, 2018. (Cited on page 13.)

[Fle18a] Lukas Fleischer. Efficient Membership Testing for Pseudovarieties of Finite Semigroups. *CoRR*, abs/1805.00650, 2018. (Cited on page 12.)

[Fle18b] Lukas Fleischer. The Intersection Problem for Finite Semigroups. In *DLT 2018, Proceedings*, volume 11088 of *LNCS*, pages 318–329. Springer, 2018. (Cited on pages 11, 12, 50, 60, and 78.)

[Fle18c] Lukas Fleischer. On the Complexity of the Cayley Semigroup Membership Problem. In *CCC 2018, Proceedings*, volume 102 of *LIPIcs*, pages 25:1–25:12. Dagstuhl Publishing, 2018. (Cited on pages 11, 12, 50, and 65.)

[Ger70] J. A. Gerhard. The lattice of equational classes of idempotent semigroups. *Journal of Algebra*, 15:195–224, 1970. (Cited on page 21.)

[GKM06] Anna Gál, Michal Koucký, and Pierre McKenzie. Incremental branching programs. In *CSR 2016, Proceedings*, pages 178–190. Springer, 2006. (Cited on page 64.)

[Has86] J Hastad. Almost optimal lower bounds for small depth circuits. In *STOC 1986, Proceedings*, pages 6–20. ACM, 1986. (Cited on page 32.)

[HK04] Markus Holzer and Barbara König. On deterministic finite automata and syntactic monoid size. *Theoretical Computer Science*, 327(3):319–347, November 2004. (Cited on page 29.)

[HK11] Markus Holzer and Martin Kutrib. Descriptional and computational complexity of finite automata — a survey. *Inf. Comput.*, 209(3):456–470, 2011. (Cited on page 78.)

[JL76] Neil D. Jones and William T. Laaser. Complete problems for deterministic polynomial time. *Theoretical Computer Science*, 3(1):105–117, 1976. (Cited on page 64.)

[JLL76] Neil D. Jones, Y. Edmund Lien, and William T. Laaser. New problems complete for nondeterministic log space. *Mathematical Systems Theory*, 10(1):1–17, Dec 1976. (Cited on pages 11, 64, and 67.)

[JM06] Marcel Jackson and Ralph McKenzie. Interpreting graph colorability in finite semigroups. *International Journal of Algebra and Computation*, 16(01):119–140, 2006. (Cited on page 87.)

[JR91] Tao Jiang and Bala Ravikumar. Minimal NFA problems are hard. In *ICALP 1991, Proceedings*, volume 510 of *LNCS*, pages 629–640. Springer, 1991. (Cited on pages 64 and 78.)

[JV10] Marcel Jackson and Mikhail Volkov. *The Algebra of Adjacency Patterns: Rees Matrix Semigroups with Reversion*, pages 414–443. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. (Cited on page 87.)

[Kle56] Steven C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, number 34 in Annals of Mathematics Studies, pages 3–40. Princeton University Press, 1956. (Cited on page 9.)

[Koz77] Dexter Kozen. Lower bounds for natural proof systems. In *FOCS 1977, Proceedings*, pages 254–266, Providence, Rhode Island, 1977. IEEE Computer Society Press. (Cited on pages 64 and 78.)

[KS95] O. G. Kharlampovich and M. V. Sapir. Algorithmic problems in varieties. *International Journal of Algebra and Computation*, 05(04n05):379–602, 1995. (Cited on page 87.)

[MC87] Pierre McKenzie and Stephen A. Cook. The parallel complexity of abelian permutation group problems. *SIAM J. Comput.*, 16(5):880–909, October 1987. (Cited on page 65.)

[MS72] Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *13th Annual Symposium on Switching and Automata Theory*, pages 125–129. IEEE Computer Society, 1972. (Cited on page 10.)

[Pet77] M. Petrich. *Lectures in Semigroups.* Wiley, 1977. (Cited on page 21.)

[Pin86] Jean-Éric Pin. *Varieties of Formal Languages.* North Oxford Academic, London, 1986. (Cited on pages 17 and 18.)

[PZ18] Thomas Place and Marc Zeitoun. The complexity of separation for levels in concatenation hierarchies. *CoRR*, abs/1810.09287, 2018. (Cited on page 94.)

[Rei82] Jan Reiterman. The Birkhoff theorem for finite algebras. *Algebra Universalis*, 14:1–10, 1982. (Cited on page 20.)

[Rei08] Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4):17:1–17:24, September 2008. (Cited on pages 34, 65, and 89.)

[Rho99] John Rhodes. Undecidability, automata, and pseudovarities of finite semigroups. *International Journal of Algebra and Computation*, 09(03n04):455–473, 1999. (Cited on pages 40 and 87.)

[Rot99] J. Rotman. *An Introduction to the Theory of Groups.* Graduate Texts in Mathematics. Springer New York, 1999. (Cited on page 20.)

[RS09] John L. Rhodes and Benjamin Steinberg. *The $\mathfrak{q}$-theory of finite semigroups.* Springer Monographs in Mathematics. Springer, 2009. (Cited on pages 17, 18, 20, 25, 88, and 89.)

[Sch65] Marcel-Paul Schützenberger. On finite monoids having only trivial subgroups. *Inf. Control*, 8:190–194, 1965. (Cited on page 9.)

[Sim68] C. C. Sims. Computational methods in the study of permutation groups. In *Proceedings of the Conference on Computational Problems in Abstract Algebra 1967, Oxford, United Kingdom*, pages 169–183, New York, 1968. Pergamon. (Cited on page 64.)

[SW15] Howard Straubing and Pascal Weil. Varieties. *CoRR*, abs/1502.03951, 2015. (Cited on page 87.)

[Tis80] A. V. Tishchenko. The finiteness of a base of identities for five-element monoids. *Semigroup Forum*, 20(1):171–186, Dec 1980. (Cited on page 24.)

[Tra81] A. N. Trahtman. A basis of identities of the five-element brandt semigroup. *Ural. Gos. Univ. Mat. Zap*, 12(3):147–149, 1981. (Cited on page 24.)

[Ver] A. S. Vernitkii. Towards a description of classes of algebras with the polynomial membership problem. Unpublished. (Cited on page 87.)

[Vol95] M.V. Volkov. On a class of semigroup pseudovarieties without finite pseudoidentity basis. *International Journal of Algebra and Computation*, 05(02):127–135, 1995. (Cited on pages 25 and 87.)

[Vol97] Mikhail Volkov. Conditional equations for pseudovarieties. Technical report, Department of Mathematics and Mechanics, Ural State University, 1997. (Cited on page 87.)

[Vol99] Heribert Vollmer. *Introduction to Circuit Complexity.* Springer, Berlin, 1999. (Cited on page 32.)

[Wil06] John S. Wilson. Finite axiomatization of finite soluble groups. *Journal of the London Mathematical Society*, 74(3):566–582, 2006. (Cited on page 40.)

[Yam67] Miyuki Yamada. Regular semi-groups whose idempotents satisfy permutation identities. *Pacific J. Math.*, 21(2):371–392, 1967. (Cited on page 22.)

[Yam13] Tomoyuki Yamakami. Uniform-circuit and logarithmic-space approximations of refined combinatorial optimization problems. In *COCOA 2013, Proceedings*, pages 318–329, Cham, 2013. Springer. (Cited on page 64.)

[Yao85] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles. In *SFCS 1985, Proceedings*, pages 1–10. IEEE Computer Society, 1985. (Cited on page 32.)