RATIONAL MONOID AND SEMIGROUP AUTOMATA

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2010

Elaine L. Render
School of Mathematics

Contents

Word count 38527

List of Tables

List of Figures

The University of Manchester

Elaine L. Render
Doctor of Philosophy
Rational Monoid and Semigroup Automata
June 28, 2010

We consider a natural extension to the definition of M-automata which allows the automaton to make use of more of the structure of the monoid M, and by removing the reliance on an identity element, allows the definition of S-automata for S an arbitrary semigroup. In the case of monoids, the resulting automata are equivalent to *valence automata with rational target sets* which arise in the theory of regulated rewriting. We focus on the polycyclic monoids, and show that for polycyclic monoids of rank 2 or more they accept precisely the context-free languages. The case of the bicyclic monoid is also considered. In the process we prove a number of interesting results about rational subsets in polycyclic monoids; as a consequence we prove the decidability of the rational subset membership problem, and the closure of the class of rational subsets under intersection and complement. In the case of semigroups, we consider the important class of completely simple and completely 0-simple semigroups, obtaining a complete characterisation of the classes of languages corresponding to such semigroups, in terms of their maximal subgroups. In the process, we obtain a number of interesting results about rational subsets of Rees matrix semigroups.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright in it (the "Copyright") and s/he has given The University of Manchester the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.

ii. Copies of this thesis, either in full or in extracts, may be made **only** in accordance with the regulations of the John Rylands University Library of Manchester. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.

iii. The ownership of any patents, designs, trade marks and any and all other intellectual property rights except for the Copyright (the "Intellectual Property Rights") and any reproductions of copyright works, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and exploitation of this thesis, the Copyright and any Intellectual Property Rights and/or Reproductions described in it may take place is available from the Head of the School of Mathematics.

Acknowledgements

I would like to thank Mark Kambites, whose meticulous attention to detail has hopefully rubbed off on me; Sasha Borovik, without whom I would probably not be in this position today and my mother, for putting up with me being a poverty stricken student for so many years longer than most.

Chapter 1

Introduction

The effectiveness of algebraic methods in classical automata theory is long established. For example, for a given finite automaton there exists an associated semigroup whose structure completely encapsulates the action of the automaton, the so called syntactic semigroup.

By adding a memory register to a classical finite automaton, the accepting power can be increased. Language families such as the context-free languages may be defined in this way. G-automata are finite automata augmented with a memory register which may at any time contain an element of a given group G. Computation in the memory register takes the form of right multiplication by elements of G; the identity element of the group defines the accepting configuration of the register, allowing us to think about G-automata languages. This is a natural algebraic generalisation of the memory registers appearing in the definition of automata such as those accepting the context-free languages.

It turns out that many formal language classes may be redefined using G-automata, providing a unifying approach to classical language families generated by automata with memory storage such as the context-free languages [22] and the counter languages [40]. This reinterpretation of disparate memory structures and their actions in an algebraic framework has allowed results and techniques from algebra to aid in new discoveries in formal language theory.

One particular area of interest in combinatorial group theory is the subject of

decision problems, such as deciding whether a given group element is equivalent to the identity element of the group. The rational subset problem, that is, the problem of deciding if a given word belongs to a given rational subset (or equivalently, if it is accepted by a given automaton) generalises a number of interesting decision problems such as the word problem. G-automata have helped provide new related results, see for example [18, 23, 34].

We may also consider M-automata where M is a monoid, rather than a group. M-automata are closely related to regulated rewriting systems, and in particular the *valence grammars* introduced by Pãun [45]: the languages accepted by M-automata are exactly the languages generated by regular M-valence grammars [20].

While M-automata appear at first sight to provide much more flexibility than their group counterparts, the extent to which such an automaton can fully utilise the structure of the register monoid is somewhat limited. Indeed, if the register ever contains an element of a proper ideal, then no sequence of actions of the automaton can cause it to contain the identity again; thus, the automaton has entered a "fail" state from which it can never accept a word. It follows that the automaton can make effective use of only that part of the monoid which does not lie in a proper ideal.

A natural way to circumvent this weakness is to weaken the requirement that the identity element be the sole accepting configuration of the register, and instead permit a more general set of initial and terminal configurations. Permitting more general terminal sets was first suggested in [22], and has recently reappeared in the study of regulated rewriting systems, where the introduction of *valence grammars with target sets* leads naturally to a corresponding notion of a *valence automaton with target set* [19, 20].

If we are to retain the advantages of monoid automata, as an elegant and easily manipulated way of describing important language classes, it is clearly necessary to place some kind of restriction on the class of subsets permitted for initial and terminal configurations. Obvious choices include the finite subsets or the finitely generated submonoids, but from a computational perspective, the most natural choice seems to be the more general *rational subsets* of the monoid. These sets, which have

been the subject of intensive study by both mathematicians and computer scientists (see for example [4, 37, 47, 52, 54]), are general enough to significantly add to the power of monoid automata, while remaining sufficiently well-behaved to permit the development of a meaningful theory.

The main objective of this thesis is to lay the foundations for the systematic study of monoid automata with rational initial and accepting sets.

Since the introduction of more general initial and terminal sets removes the special role played by the identity element we are able to consider automata with an S-register where S is an arbitrary semigroup perhaps without an identity element. We believe it may be possible to extend even further the success of monoid automata as an elegant algebraic description of important language classes, and to use them to study the structure of more general semigroups.

The rest of this thesis is arranged as follows: in Chapter 2 we recall some elementary definitions from semigroup theory and introduce finite automata. The properties of languages accepted by finite automata are explored in detail, for automata defined over the free monoid (yielding the regular languages), and for finite automata defined over more general semigroups. Grammars as a tool for language generation are introduced, including context-free grammars and regulated grammars, which are closely linked to M-automata. After a brief consideration of the links between the decision problems of combinatorial group and semigroup theory and formal language theory, we define notions of grouping for languages, culminating in a discussion of the classical Chomsky hierarchy and the language families usually included within it.

In Chapter 3 we collect together in a cohesive form results from the literature relating to M-automata for M taken from specific families of groups and monoids. Finite, cyclic and commutative monoids and groups are considered first, including a number of results connecting the word problems of such groups with M-automata defined over them. We next consider the free groups, which are closely linked to the context-free languages. Similarly connected are the polycyclic monoids; an important result of Chomsky and Schutzenberger concerning context-free languages may be reinterpreted using free groups and polycyclic monoids. Lastly in this chapter we

explore the possibilities for nilpotent groups, giving a simple example which demonstrates their potential as an interesting class of groups for study in an M-automaton context.

In Chapter 4 we look at the structural properties of monoids such as ideals and zero elements, and consider *simple*, *0-simple*, *completely simple* and *completely 0-simple* monoids. We prove a number of results concerning the resulting limitations on the functioning of M-automata defined over monoids with these properties. The first part of the chapter culminates in a result analogous to an important result of Mitrana and Stiebe [40] which appears in Chapter 3, and a result outlining the potential properties of the class of languages accepted by M-automata for a given monoid M. We next introduce rational M-automata, the extended definition of M-automata discussed above, and consider some foundational properties of these automata with respect to monoid structure.

In Chapter 5 we explore our extended definition of M-automata for monoids taken from the important class of *polycyclic monoids*. The polycyclic monoid of rank n is the natural algebraic model of a pushdown store on an n letter alphabet. For M a polycyclic monoid of rank 2 or more, it is well known that M-automata are equivalent to pushdown automata, and hence that the languages accepted are precisely the context-free languages. The polycyclic monoid of rank 1 is called the *bicyclic monoid*, and as we shall have seen in Chapter 3 bicyclic monoid automata accept precisely the *partially blind one-counter languages* as defined by Greibach [26]. We first study the structure of rational subsets in polycyclic monoids, and then use these results to prove the relationship between rational polycyclic monoid automata languages and the context-free languages.

In Chapter 6 we consider *completely 0-simple* semigroups. Semigroups of this type may be characterised using *Rees matrix semigroups* constructed from groups. These constructions play a crucial role in the structure theory of semigroups, making them an interesting candidate for study in the context of our extended S-automaton definition. We first study the relationship between rational subsets and the Rees matrix construction, in the process proving a number of results about the structure of

rational subsets of completely 0-simple semigroups. We then go on to give a complete description of the classes of language accepted by rational S-automata where S is a completely simple or completely 0-simple semigroup.

Chapter 2

Preliminaries

In this chapter we introduce some fundamental algebraic and language theoretic definitions and results which will be the basis for the rest of this thesis.

2.1 Algebraic notions

We begin by introducing some algebraic notions. A *binary operation* on a set S is a mapping which takes ordered pairs of elements of S to single elements of S:

$$f : S \times S \to S.$$

We usually write this $a \cdot b = c$, and in fact when the operation in question is clear, the dot will be omitted. Such a binary operation is said to be *associative* if for all $a, b, c \in S$,

$$(a \cdot b) \cdot c = a \cdot (b \cdot c).$$

A *semigroup* S is a set together with an associative binary operation. An element $e \in S$ is called a *neutral* or *identity* element of the semigroup if for all $a \in S$,

$$ae = ea = e.$$

A semigroup S with a neutral element is called a *monoid*. We will usually denote such an identity element by 1.

An element $a \in S$ is said to have an *inverse* element, denoted a^{-1}, if

$$aa^{-1} = a^{-1}a = e$$

where e is the identity element of our monoid. A monoid in which every element has an inverse is called a *group*.

Let T, U be subsets of a semigroup S. We extend the definition of multiplication in the semigroup to subsets as follows.

$$T \cdot U = \{t \cdot u \mid t \in T, u \in U\}$$

where $\cdot$ denotes the associative binary operation in the semigroup. As in the case of individual elements, we conventionally will not include the dot, and note that multiplication of subsets is also associative.

A *subsemigroup* S' of a semigroup S is a subset of S which is closed under the associative binary operation of S. A *submonoid* S' of a semigroup S is a subset of S which is closed under the binary operation of S and contains an element $e \in S'$ which behaves as an identity element in S'. That is, S' is a monoid. A *subgroup* of a semigroup S is a subset S' of S which is itself a group. We use the notation $S \geq S'$ or for S' a *proper* subgroup $(S' \neq S)$ $S > S'$.

For a subgroup H of a group G a *left coset* of H in G is a subset of the form $gH = \{gh \mid h \in H\}$ and a *right coset* is one of the form $Hg = \{hg \mid h \in H\}$ for some $g \in G$. The cardinality of the set of distinct left cosets is always equal to the cardinality of the set of distinct right cosets for any given subgroup H. This number is called the *index* of the subgroup H in G. A subgroup N of G is called *normal* if for all $n \in N$ and $g \in G$, $gng^{-1} \in N$. An important example of a normal subgroup is the *centre* of G, defined

$$Z(G) = \{z \in G \mid zg = gz \ \forall g \in G\},$$

the set of elements which *commute* with every element of G.

A binary relation $\sim$ on a set S is simply a collection of ordered pairs of the form $(a, b) \in S \times S$. If the pair (a, b) is in our relation $\sim$ then we may write $a \sim b$, "a is $\sim$ related to b". Given a binary relation $\sim$ on a semigroup S we say that $\sim$ is a *congruence relation*, or simply *congruence*, if it satisfies the following four properties.

(i) For all $a \in S$, $a \sim a$ (*reflexivity*);

(ii) For all $a, b \in S$, if $a \sim b$ then $b \sim a$ (*symmetry*);

(iii) For all $a, b, c \in S$, if $a \sim b$ and $b \sim c$ then $a \sim c$ (*transitivity*);

(iv) For all $a, a', b, b' \in S$, if $a \sim a'$ and $b \sim b'$ then $ab \sim a'b'$ (*compatibility*).

A binary relation satisfying the first three conditions is called an *equivalence relation*. Every relation on a semigroup S (that is, every subset of $S \times S$) is contained in a unique minimal congruence on S, called the *congruence generated by* the relation.

Given two relations $R \subseteq X \times Y$ and $S \subseteq Y \times Z$ over sets X, Y and Z, the *composition* of R and S is the set

$$R \circ S = \{(x, z) \mid \exists y \in Y : (x, y) \in R \wedge (y, z) \in S\} \subseteq X \times Z.$$

A *semigroup (homo)morphism* is a mapping from one semigroup into another which respects the operations of the two semigroups, that is, $\phi : S \to S'$ where S and S' are semigroups and where $(a\phi)(b\phi) = (ab)\phi$ for all $a, b \in S$ (the convention throughout will be to apply maps on the right). We denote by $S\phi$ the image of the whole of S under the morphism ϕ, that is, the set $\{s' \in S' \mid s' = s\phi \text{ for some } s \in S\}$. We say that $S\phi$ is a *homomorphic image* of S. For an element $s' \in S'$ we call an element $s \in S$ such that $s\phi = s'$ an *inverse image* of s' and write $s'\phi^{-1}$ for the set of all inverse images of s'. An injective and surjective homomorphism is called an *isomorphism*. If there exists an isomorphism between two semigroups S and S' we say that they are *isomorphic*, denoted $S \cong S'$.

For a given congruence $\sim$ the equivalence classes induced form a semigroup with multiplication defined by

$$[a][b] = [ab]$$

where $[a]$ denotes the equivalence class containing a. The semigroup defined in this way is denoted $S/\sim$. The map $a \mapsto [a]$ is a surjective morphism from S onto $S/\sim$.

One of the most natural types of semigroup, monoid or group in terms of its structure is a *free* one. In full generality we have the following definition. Let F be an algebra in a class $\mathscr{C}$ of algebras. Then F is *free* in $\mathscr{C}$ if there is a subset

$X \subseteq F$ such that every function from X to an algebra $M \in \mathscr{C}$ extends uniquely to a morphism from F to M.

Thinking in terms of the types of structures we shall encounter, let A be a finite alphabet of symbols. Then we denote by A^* the *free monoid* on A, and by A^+ the *free semigroup* on A. The *free group* on A is denoted F_A. A more intuitive definition of free objects will follow in Section 2.1 below.

For a monoid or group M, we call a surjective morphism $\sigma : X^* \to M$ from a free monoid X^* to the monoid M a *choice of generators* for M, and the elements of the set X the *generators* of M. The choice of generators is called *finite* if X is finite.

A *presentation* for a monoid M is of the form

$$\langle X \mid R \rangle$$

where X is a set of generators, and $R \subseteq X^* \times X^*$. The monoid M is then derived from the presentation as $M = X^*/\sim$ where $\sim$ is the smallest congruence containing the relations in R. Since the map $X^* \to M$ is a surjective morphism it is a choice of generators for M. The presentation is called *finite* if A and R are finite.

For general semigroups a *choice of generators* is a surjective morphism $\sigma : A^+ \to S$ from the free semigroup A^+ to S. Again we refer to elements of the set A as *generators* of S. A semigroup *presentation* for a semigroup S takes the form $\langle A \mid R \rangle$ where A is a generating set for S and the semigroup is as before derived from the presentation as $A^+/\sim$ where $\sim$ is the smallest congruence containing the relations $R \subseteq A^+ \times A^+$.

A monoid or semigroup is said to be *finitely generated* if it admits a finite choice of generators, and *finitely presented* if it is isomorphic to the monoid derived from a finite presentation.

For two semigroups S and S' there are many ways to construct new semigroups from them. The one which will be most useful throughout this thesis will be the *direct product*:

$$S \times S' = \{(s, s') \mid s \in S,\ s' \in S'\}.$$

The direct product of two semigroups is a semigroup itself under the operation

$$(s, s')(t, t') = (st, s't')$$

and is naturally generated by the set $\{X \cup X'\}$ where X and X' are generating sets for the semigroups S and S' respectively.

Another way we may wish to produce new semigroups from existing ones is to *adjoin* new elements with specific interesting properties. Since the existence of an identity element in a semigroup often makes calculations more straightforward, we begin by considering adjoining an identity element. Let S be a semigroup. We denote by S^1 the semigroup obtained from S by adjoining an identity element 1, where

$$S^1 = \begin{cases} S & \text{If } S \text{ contains an identity element,} \\ S \cup \{1\} & \text{otherwise.} \end{cases}$$

We extend the multiplication of S to S^1 in the unique way which makes 1 an identity element.

Another interesting type of element which we may wish to adjoin to a semigroup is a *zero*. For S a semigroup we call an element $0 \in S$ a *zero element* if for all $x \in S$ we have

$$0x = x0 = 0$$

and define S^0, the *semigroup with zero* to be

$$S^0 = \begin{cases} S \cup \{0\} & \text{if } S \text{ has no zero element,} \\ S & \text{otherwise} \end{cases}$$

with multiplication defined by

$$st = \begin{cases} s \cdot t & \text{If } s, t \in S, s, t \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

A useful way of considering the structure of a semigroup is using *Green's relations* [10]. We say that two elements $a, b \in S$ are $\mathscr{L}$*-related*, written $a \mathscr{L} b$ if and only if $S^1a = S^1b$. Similarly we say that a and b are $\mathscr{R}$*-related*, written $a \mathscr{R} b$ if and only if $aS^1 = bS^1$. If for elements $a, b \in S$ we have $S^1aS^1 = S^1bS^1$ we say that a and b are $\mathscr{J}$*-related*, written $a \mathscr{J} b$. We call an equivalence class of $\mathscr{L}$-related elements an $\mathscr{L}$*-class*, an equivalence class of $\mathscr{R}$-related elements is called an $\mathscr{R}$*-class* and an equivalence class of $\mathscr{J}$-related elements is called a $\mathscr{J}$*-class*. For a given element

$a \in S$, we denote the $\mathscr{L}$-class containing a by $\mathscr{L}_a$, and the $\mathscr{R}$-class containing a by $\mathscr{R}_a$.

Proposition 2.1.1 ([33]). *The relations $\mathscr{L}$ and $\mathscr{R}$ commute.*

Proof. Let $a, b \in S$ and assume that $(a, b) \in \mathscr{L} \circ \mathscr{R}$. Then there exists some $c \in S$ such that $a\mathscr{L}c$ and $c\mathscr{R}b$. Hence there exist elements $x, , y, u, v \in S$ such that

$$xa = c \quad cu = b$$

$$yc = a \quad bv = c.$$

Let $d = ycu \in S$. Then

$$au = ycu = d \; dv = ycuv = ybv = yc = a$$

and we may conclude that $a\mathscr{R}d$. Similarly

$$yb = ycu = d \; xd = xycu = xau = cu = b$$

and $d\mathscr{L}b$. Therefore $\mathscr{L} \circ \mathscr{R} \subseteq \mathscr{R} \circ \mathscr{L}$. The other direction is proved similarly.

□

The relation $\mathscr{D}$ is the join of the relations $\mathscr{L}$ and $\mathscr{R}$. Since $\mathscr{L}$ and $\mathscr{R}$ commute it is the smallest equivalence relation containing both $\mathscr{L}$ and $\mathscr{R}$. An equivalence class of $\mathscr{D}$-related elements is called a $\mathscr{D}$-class. We define the $\mathscr{H}$ relation as $\mathscr{H}$=$\mathscr{L} \cap \mathscr{R}$, the intersection of the $\mathscr{L}$ and $\mathscr{R}$ relations. An equivalence class of $\mathscr{H}$-related elements is called an $\mathscr{H}$-class.

An *ideal* I of a semigroup S is a subset I of S with the property that $S^1IS^1 \subseteq I$. Notice in particular that an ideal is a subsemigroup. We say that an ideal is *proper* if it is properly contained in the semigroup S ($I \neq S$). To each ideal I is associated a congruence ρ_I on S such that $(s, t) \in \rho_I$ if and only if either $s, t \in I$ or $s = t$. The quotient monoid, usually denoted S/I , is called a *Rees quotient*, and takes the form

$$S/I = \{I\} \cup \{\{x\} \mid x \in S \setminus I\},$$

though it is isomorphic to $S \setminus I \cup \{0\}$ with the binary operation defined

$$st = \begin{cases} s \cdot t & \text{If } s, t, s \cdot t \in S \setminus I \\ 0 & \text{otherwise} \end{cases}$$

where $\cdot$ is the binary operation of the original semigroup S. It is most convenient to consider it in this way.

The free monoid

The monoids which will feature most prominently in this thesis will be the finitely generated free monoids. It is these structures which form the basis for all of formal language theory. In this section we introduce some related definitions.

Let Σ be a finite alphabet of symbols. Then we denote by Σ^* the set of all finite strings of symbols from Σ and by ϵ the empty string. We call such strings *words*. Under the operation of concatenation and with the neutral element ϵ, Σ^* forms a free monoid. We refer to ϵ as the *empty word*. We denote by $|w|$ the *length* of a given word and by $|w|_a$ the number of occurrences of some given letter $a \in \Sigma$ in the word. A word $u \in \Sigma^*$ is said to be a *factor* of a word $w \in \Sigma^*$ if there exist words $v, z \in \Sigma^*$ such that $w = vuz$. If we can choose $v = \epsilon$ we say that u is a *left factor* of w; if we can choose $z = \epsilon$ we say that u is a *right factor* of w.

2.2 Finite automata

Next, we introduce some basic ideas from formal language theory; we begin with *finite automata*.

Finite automata and regular languages

The most intuitive way to define finite automata is using *graphs*. A finite graph is a tuple (V, E) where V is a finite set of *vertices* and E is a finite set of *edges* connecting certain vertices together; each edge is a two element subset of V. A *directed* graph is a graph where each edge is endowed with a direction (that is, an edge is considered to

start at one vertex and end at another). A *finite automaton* over Σ^* is a finite directed graph with each edge labelled by an element of Σ or by ϵ, and with a distinguished initial vertex and a set of distinguished terminal vertices. In the sequel vertices will be referred to as *states*. A word $w \in \Sigma^*$ is *accepted by* the automaton if there exists a sequence of consecutive edges (a *path*), connecting the initial state with some terminal state labelled cumulatively with w. That is, there exists a path with edges labelled $w_1, \ldots, w_n$ for some $n \in \mathbb{N}$ with $w_1 w_2 \ldots w_n = w$. The set of all words accepted by the automaton is often denoted L or for an automaton A sometimes $L(A)$, and is called the *language* accepted by A. Such a language is called *rational* or *regular*.

A finite automaton as defined above is called *deterministic* if no edges are labelled by ϵ and for each $a \in \Sigma$ and for each state q in the automaton there exists at most one edge starting at q labelled by a. If this is not the case we say that the automaton is *non-deterministic*. In the case of regular languages, we may always find a deterministic finite automaton accepting the same language as a given non-deterministic automaton [32].

We refer to edges in a finite automaton which have label ϵ as *ϵ-transitions*. Note that in the case of regular languages, if there exists a finite automaton accepting the language which includes ϵ-transitions we may always find another automaton accepting precisely the same language which contains no ϵ-transitions [32]. For a given edge from a state p to a state q it will be useful to refer to p as the *source* state of the edge, and q as the *target* state of the edge.

It is reasonable to consider automata with edges labelled by words $w \in \Sigma^*$ rather than simply letters from Σ. However, usually it will be more convenient to use the latter labelling since they are equivalent. Indeed, consider an automaton with edges labelled from Σ^*. Then an edge labelled by $w \in \Sigma^*$ with $w = w_1 \ldots w_n$ (for $w_i \in \Sigma$, $i = 1, \ldots, n$) may be split into n consecutive edges, each labelled by w_i for $i = 1, \ldots, n$.

An obvious question to ask is whether the condition that there be a unique initial state is necessary. We define a *generalised* finite automaton to be a finite automaton with a set of distinguished initial states. Then a word $w \in \Sigma^*$ is accepted by A if

there exists a path labelled by w connecting an initial state q to a terminal state q'. We shall see below (Proposition 2.2.4) that this generalisation adds no extra power to the automaton.

We define the notation A^* for a set A to be the set of all possible strings consisting of the concatenation of zero or more words from A. For example, let $A = \{10, 11\}$, then

$$\{10, 11\}^* = \{\epsilon, 10, 11, 1011, 1110, 1010, 1111, \ldots\}$$

is the submonoid generated by A. We call this operation the *Kleene star*. Note that this use of the $*$ notation is in line with our previous use to define a free monoid. We also use the notation A^+, which denotes the set of all possible strings which are the concatenation of one or more words from A, that is, $A^+ = A^* \setminus \epsilon$ (where $\epsilon \notin A$). The *complement* of a language $L \subseteq \Sigma^*$ is the set $\Sigma^* \setminus L$ of all strings over the alphabet Σ which do not appear in L.

The regular languages are equivalent in expressive power to languages built from *regular expressions* [32]. Such expressions are defined inductively as follows.

- $\emptyset$ is a regular expression and denotes the empty set.
- The empty word ϵ, and each $a \in \Sigma$ is a regular expression denoted $\{\epsilon\}$ and $\{a\}$ respectively.
- If E_1, E_2 are regular expressions then so are $E_1 \cup E_2$ and $E_1 E_2$.
- If E_1 is a regular expression then so is E_1^*.

For a regular expression E we write $L(E)$ for the language denoted by E.

We say that a property is *testable* if there exists some finite terminating algorithm which decides if the property is satisfied by a given structure.

Proposition 2.2.1 ([32])**.** *Emptiness of regular languages is testable, that is, there exists an algorithm which, given as input a finite automaton, decides if the language which it accepts is empty.*

A useful tool for showing that a given language is not regular is the so called *pumping lemma for regular sets*. It says that, given a sufficiently long word in a regular language, we may find a subword conforming to certain properties which may be "pumped", that is, repeated any number of times, and the resulting word will still be contained in the original language.

Lemma 2.2.2 (The Pumping Lemma for Regular Languages, [32]). *Let $L \subseteq \Sigma^*$ be a regular language. Then there exists a constant $n \in \mathbb{N}$ such that if $z \in \Sigma^*$ is any word in L with $|z| \geq n$ we may write $z = uvw$ such that*

- $|uv| \leq n$,
- $|v| \geq 1$ *and*
- *for all $i \geq 0$, $uv^i w \in L$.*

Furthermore, n may be chosen to be no greater than the number of states in the smallest finite automaton accepting L.

A subset S of a monoid M is said to be *recognisable* if there exists a finite monoid N, a homomorphism $\phi : M \to N$ and a subset T of N such that $S = T\phi^{-1}$. When the monoid M is taken to be the free monoid Σ^* on a finite alphabet Σ the set of recognisable subsets is exactly the set of regular languages, that is, the rational subsets of Σ^*. This result is known as Kleene's Theorem.

Theorem 2.2.3 (Kleene's Theorem, [32]). *Let Σ be a finite alphabet. The recognisable subsets of Σ^* are exactly the regular languages.*

Finite automata over more general semigroups

We now shift our focus from the free monoid to semigroups in general. Let S be a semigroup. Then the set of rational subsets of S is defined to be the closure of the set of finite subsets of S under union, subset multiplication (and hence concatenation) and generation of submonoids. An alternative and equivalent definition can be given in terms of finite automata.

If S is a semigroup then a *finite automaton over* S is a finite directed graph with each edge labelled by an element of S, and with a distinguished initial state and a set of distinguished terminal states. An element $s \in S$ is accepted by the automaton if there exists some path connecting the initial state with some terminal state labelled cumulatively with s. That is, there exists a path with edges labelled $s_1, \ldots, s_n$ for some $n \in \mathbb{N}$ with $s_1 \cdot s_2 \cdot \ldots \cdot s_n = s$ where $\cdot$ denotes the operation in the semigroup S. The *subset accepted* is the set of all elements accepted; a subset of S is accepted by a finite automaton precisely when it is a *rational subset* of S as defined above. The rational subsets of Σ are the regular languages and the rational subsets of a general semigroup S are the homomorphic images in S of regular languages.

It should be noted that rational subsets of semigroups are not as well behaved as languages over the free monoid. Some of the concepts discussed in the previous section, such as determinism, do not make sense in this more general setting.

We extend the definition of a generalised finite automaton presented previously to semigroups as follows: A *generalised finite automaton* over a semigroup S is a finite automaton defined over S which, instead of a single unique initial state, may have some set of initial states $I \subseteq Q$ where Q is the state set.

Proposition 2.2.4. *Let $L \subseteq S$ be a subset of the semigroup S, accepted by a generalised finite automaton. Then L is rational.*

Proof. Let A be a generalised finite automaton such that $L(A) = L$ and let $I \subseteq Q$ be the set of initial states where Q is the state set of A.

Let B be an identical copy of A. We add a new state q_s to B which we designate as the unique initial state. For each edge connecting some $q \in I$ to some state q' labelled by a we add an edge labelled by a connecting q_s to q'. We repeat this for each $q \in S$ and $a \in \Sigma$.

The resulting automaton accepts exactly the language L. □

A related result is the following.

Proposition 2.2.5. *Let $L, K \subseteq S$ be two rational subsets of a semigroup S. Then $L \cup K$ is also rational.*

Proof. Let A and B be two finite automata over S accepting the sets L and K respectively. By taking the set of initial states to consist of the initial state of A with the initial state of B we may view A and B as a single generalised finite automaton (albeit one with two unconnected components). By Proposition 2.2.4 there exists a finite automaton over S with a single initial state accepting the set $L \cup K$ as required. □

We note that given two rational subsets over the same semigroup S, their intersection may not necessarily again be a rational subset. Since regular languages are defined over a free monoid there exists a unique way to write any given element with respect to a specific generating set. In the case of general semigroups, this is not the case, and hence though an element $s \in S$ may appear in two rational sets $R, R' \subseteq S$, it may appear differently, and hence the letter by letter comparison of the words as they appear in the automata which is implied in the intersection construction for the regular case may result in a conclusion of inequality.

Recall Kleene's theorem from the previous section. Though Kleene's theorem does not apply in full generality for semigroups, there are many examples of semigroups and monoids which do satisfy an analogue of Kleene's theorem. We call such semigroups *Kleene semigroups*, or *Kleene monoids* in the case of monoids. A complete characterisation of the class of Kleene monoids has not yet been found, but many attempts have been made. Examples of classes of Kleene monoids include the Amar-Putzolu monoids [1] and small overlap monoids [36].

Both Amar-Putzolu monoids and small overlap monoids fall into the class of *rational monoids* [52]. Monoids of this type have multiplication which is in some sense "simple". We may describe a monoid M using its generating set X and its surjective choice of generators map $\sigma : X^* \to M$. Clearly there may be a number of elements $x \in X^*$ which are mapped to a given element $m \in M$. By choosing one unique such x to be the representative of m in X^*, we may construct a map from X^* to itself. Then a monoid M is rational if there exists a function constructed in this way which is *rational*, that is, it is a rational relation which is functional (see Section

2.2 for more on rational relations).

As it turns out, all rational monoids are Kleene [52]. The converse however is not true [46].

We require the following result about rational subsets of groups, which is well known.

Proposition 2.2.6. *Let G be a group. If $X \subseteq G$ is rational then the subset $X^{-1} = \{x^{-1} \mid x \in X\}$ is also rational.*

Proof. Let $X \subseteq G$ be a rational subset of a group G. Then X is accepted by some finite automaton A. We construct a new generalised automaton B with

- state set Q where Q was the state set of A,
- initial state set F where F was the set of terminal states of A,
- unique terminal state q_0 where q_0 was the initial state of A and
- for each edge from state p to q labelled by $g \in G$ in A an edge from q to p in B labelled by $g^{-1} \in G$.

It is clear that the resulting automaton accepts exactly the set X^{-1} and so by Proposition 2.2.4, X^{-1} is rational. □

Rational transductions and homomorphisms

We begin by defining rational relations. Relations, and by extension, transductions, are a useful tool for showing the inclusion of languages in certain language classes.

Let Ω and Σ be finite alphabets, and consider a finite automaton over the direct product $\Omega^+ \times \Sigma^*$; the subset R of $\Omega^+ \times \Sigma^*$ that it recognizes is called a *rational relation*. Hence a rational relation is simply a rational subset of the direct product of the given free semigroups or monoids. The *image* of a language $L \subseteq \Omega^+$ under the relation R is defined to be the set of words $y \in \Sigma^*$ such that $(x, y) \in R$ for some $x \in L$.

The following theorem by Nivat gives a useful characterisation of rational relations. We note first a definition: the *projection* of $(X \cup Y)^*$ onto X^* is the morphism $\pi_X : (X \cup Y)^* \to X^*$ uniquely defined by $\pi_X(x) = x$ for $x \in X$ and $\pi_X(x') = 1$ for $x' \in Y$. We define the projection of $(X \cup Y)^*$ onto Y^* similarly.

Theorem 2.2.7 ([5]). *Let X and Y be alphabets. The following are equivalent.*

(i) *$A \subset X^* \times Y^*$ is a rational relation;*

(ii) *There exists an alphabet Z, two morphisms $\varphi : Z^* \to X^*$ and $\psi : Z^* \to Y^*$ and a regular language $K \subset Z^*$ such that*

$$A = \{(h\varphi, h\psi) \mid h \in K\};$$

If $X \cap Y = \emptyset$ then we may choose Z to be $X \cup Y$ and $\psi = \pi_X$, $\varphi = \pi_Y$.

The definition of rational relation holds also for arbitrary monoids: let M, M' be monoids. Then a finite automaton over the direct product $M \times M'$ recognises a *rational relation* $R \subseteq M \times M'$. We define the image of a set $R \subseteq M$ as for free monoids above.

A rational relation between free monoids is called a *rational transduction*. An automaton recognising a rational transduction is called a *rational transducer*. In the sequel we shall use the term 'rational transduction of X' to mean 'the image under a rational transduction of X'.

Theorem 2.2.8 ([5]). *Homomorphisms and inverse homomorphisms are examples of rational transductions. For every regular language $L \subseteq X^*$, there exists a rational transduction $\sigma \subseteq X^* \times X^*$ such that for any $K \subseteq X^*$, $K\sigma = K \cap L$.*

We may extend the results of the theorem from single elements of X^* to subsets of X^* (and Y^*) and conclude the following.

Theorem 2.2.9 ([5]). *Rational transductions preserve regular and context-free languages. That is, the image $A\rho$ of a set A under a rational transduction ρ is regular if A is regular, and is context-free if A is context-free.*

Rational transductions also have the following useful property.

Theorem 2.2.10 ([5]). *The composition of two rational transductions is again a rational transduction.*

2.3 Grammars

An important tool in the definition of useful language classes are *grammars*. The most general type of grammar is a *type-0* or *unrestricted* grammar. An unrestricted grammar is a tuple (V, T, P, S) where

- V is a finite set of *variables*;
- T is a finite set of *terminals*;
- P is a finite set of *productions*; each production is of the form $\alpha \to \beta$ where $\alpha, \beta \in (V \cup T)^*$ with $\alpha \neq \epsilon$ and
- S is a special variable called the *start symbol*.

When dealing with grammars a number of conventions allow us to represent them using just a list of productions. We use capital letters from the beginning of the alphabet to denote variables; the letter S is reserved for the start symbol. Lower-case letters from the beginning of the alphabet are used to denote terminals, and lower-case letters from the end of the alphabet are used to denote strings of terminals. Mixed strings of variables and terminals are denoted by lower-case letters from the Greek alphabet.

In order to define the language derived from a given grammar we first must define two relations, $\Rightarrow_G$ and $\Rightarrow_G^*$, between strings in $(V \cup T)^*$. If $\alpha \to \beta$ is a production of P and δ and γ are any two strings in $(V \cup T)^*$ then $\delta\alpha\gamma \Rightarrow_G \delta\beta\gamma$. That is, two strings are related by $\Rightarrow_G$ when the second is obtained from the first by one application of some production. We say that $\delta\beta\gamma$ is *directly derived* from $\delta\alpha\gamma$. The relation $\Rightarrow_G^*$ is the transitive and reflexive closure of $\Rightarrow_G$. If $\alpha \Rightarrow_G^* \beta$ we say that β is *derived* from α, hence β is a *derivation* of α. The language generated by G, denoted $L(G)$

is the set $\{w \mid w \in T^*, S \Rightarrow_G^* w\}$. Hence a string is in $L(G)$ if it consists solely of terminals and can be derived from the start symbol S. A language derived from an unrestricted grammar is called *recursively enumerable*.

If we begin with an unrestricted grammar but then insist that productions must be length increasing, that is, for every production $\alpha \rightarrow \beta$ in P we have $|\beta| \geq |\alpha|$ then we have what is called a *context sensitive grammar*. Such grammars in turn define the *context sensitive* languages, or CSLs.

In fact there also exist so called *regular* grammars, which provide an alternative characterisation of the regular languages.

Context-free languages and pushdown automata

The most relevant grammar derived language class for us will be the context-free languages, defined using *context-free grammars*. The context-free languages are important for defining programming languages and for parsing, as well as being useful for many other string processing applications.

Formally we define a context-free grammar G to be a grammar (V, T, P, S) with the condition that each production is of the form $A \rightarrow \alpha$ where A is a variable and α is a string of symbols from $(V \cup T)^*$.

Then a language L is called *context-free* if it is $L(G)$ for some context-free grammar G.

An equivalent way to define the context-free languages is by using *pushdown automata*. Formally we define a pushdown automaton to be a tuple $(Q, \Sigma, \Gamma, \delta, q_0)$ where

- Q is a finite set of states;
- Σ is the finite *input* alphabet;
- Γ is the finite *stack* alphabet, including a bottom of stack marker $\perp$;
- δ is a transition relation mapping $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma$ to finite subsets of $Q \times \Gamma^*$;
- $q_0 \in Q$ is the initial state.

Informally a pushdown automaton is a finite automaton which, as well as its usual function, has control over a *stack*. A stack is essentially a list with a 'first in, last out' access rule. We refer to adding a new element to the list as *pushing* and removing an element from the list as *popping*.

Implicitly on initialisation of a run of a pushdown automaton we add the bottom of stack marker to the bottom of the stack. Then a word $w \in \Sigma^*$ is accepted by the automaton if there exists a path from the initial state of the automaton labelled by w such that the sequence of stack operations labelling the path result in the stack containing only the bottom of stack marker after the word has been read. That is, $\delta^*(q_0, w, \perp) = (q, \perp)$ for some $q \in Q$ where δ^* is the transitive, reflexive closure of δ.

We have defined the acceptance condition of a pushdown automaton in terms of the configuration of the stack. However, an alternative manner of acceptance for pushdown automata is often used, which resembles more closely the traditional acceptance condition for finite automata (that is, we have terminal states). These two types of acceptance condition are equivalent in the sense that if a set can be accepted by empty stack by one pushdown automaton, then there exists another pushdown automaton which will accept the set by terminal state and vice versa.

A useful property of context-free languages is the satisfaction of the *pumping lemma for context-free languages*. This pumping lemma, like the one given for regular languages (Lemma 2.2.2), provides a tool for proving that a given language is not context-free.

Lemma 2.3.1 ([32]). *Let $L \subseteq \Sigma^*$ be a context-free language. Then there exists an integer $n > 0$ such that any word $z \in L$ with $|z| \geq n$ can be written as $z = uvwxy$ with substrings u, v, w, x, y such that*

- $|vx| \geq 1$,
- $|vwx| \leq n$ *and*
- $uv^iwx^iy \in L$ *for all* $i \geq 0$.

Regulated grammars

Other types of grammar particularly relevant here include regulated grammars. Such systems are often also called 'grammars with controlled derivations', since these types of grammars can take some kind of control over the productions applied in the derivation step (see [12] for a general overview). *Valence grammars* [45] are an example of regulated grammars. A valence grammar is a context-free grammar within which integer values (*valences*) are assigned to each production. A derivation is then judged to be valid or not by adding the valences in the derivation; a total of zero gives a valid derivation. This definition can then be extended to other monoids (using the identity element of the monoid as the acceptance condition). Similar is the notion of *weighted grammars* suggested by Salomaa in [53].

Formally, a (context-free) valence grammar over a monoid M is a tuple (V, T, P, S, M) where V, T, and S are defined as for a context-free grammar, and the set $P \subseteq V \times (V \cup T)^* \times M$ is a finite set of *valence rules*. For a valence rule $(A \to \alpha, m)$, the production $A \to \alpha$ is a production in the usual sense of context-free grammars, and $m \in M$ is called the *valence* of the rule. The relation $\Rightarrow$ is defined as $(w, m) \Rightarrow (w', m')$ if and only if there exists a rule $(A \to \alpha, n)$ such that $w = w_1 A w_2$, $w' = w_1 \alpha w_2$ and $m' = mn$. Then the language generated by the grammar G is $L(G) = \{w \in T^* \mid (S, 1) \Rightarrow^* (w, 1)\}$ where 1 is the identity element of the monoid M.

2.4 Decision problems for groups and semigroups

In this section we consider the relationship between formal language theory and the decision problems of combinatorial group and semigroup theory.

Let G be a group. The *word problem* for a group G with respect to a choice of generators $\sigma : X^* \to G$ is the language of all words $w \in X^*$ such that $w\sigma = 1$ in G. We denote the word problem of a group G by $WP(G)$.

In the case of monoids, the *identity language* of a monoid M with choice of generators $\sigma : X^* \to M$ is the set of words $w \in X^*$ such that $w\sigma = 1$, that is, the set

of words over the generating set of the monoid which represent the identity element in M. We use the notation $ID(M)$ for the identity language of a monoid M. In the case of direct products of monoids we consider the identity language with respect to the natural generating set.

The *rational subset membership problem* for a semigroup S is the algorithmic problem of deciding, given a rational subset of S (specified using an automaton over a fixed generating alphabet) and an element of S (specified as a word over the same generating alphabet), whether the given element belongs to the given subset. The decidability of this problem is well-known to be independent of the chosen generating set [37, Corollary 3.4]. Grunschlag [29] showed that it is a virtual property (for groups), that is, it is preserved under finite extensions and taking finite index subgroups.

In fact the rational subset membership problem is a generalisation of many interesting decision problems in combinatorial group theory; we discuss some examples. The *word problem* is the problem of deciding, given a word over the generating set of a group G, whether the word represents the identity element of the group. We note the difference between this and the definition presented previously. It should be clear from the context which definition we are referring to in the sequel. The *generalised word problem* or *subgroup membership problem* is the problem of deciding, given a finite set of elements of the group G (specified as words over a generating set), and another element $g \in G$ (specified using the same generating set), whether or not the element g is contained within the subgroup generated by our set of elements. This problem can be broadened further still by considering submonoids or subsemigroups.

Since (finitely generated) subgroups, submonoids and subsemigroups are examples of rational subsets, the rational subset membership problem is a natural generalisation. It is well known that the rational subset membership problem is decidable for free groups and for free abelian groups [3, 29].

We say that a decision problem is *uniformly decidable* if there exists some algorithm which, given some presentation for a group, produces an algorithm which can solve the decision problem for the given group.

2.5 Language families

An important focus of formal language theory is to understand the connections between the many language classes. To this end, we define particular types of language classes by their closure properties.

A *family of languages* is a collection of languages containing at least one non-empty language. An ϵ-free homomorphism is a morphism h between free monoids such that $h(a) \neq \epsilon$ for any $a \neq \epsilon$. If a family of languages is closed under ϵ-free homomorphisms, inverse homomorphisms and intersection with regular languages, we call such a family a *trio* or *faithful cone* of languages. The context sensitive languages are an example of a trio of languages.

A faithful cone closed under arbitrary morphisms is termed a *full trio* or *rational cone* of languages. The regular languages and the recursively enumerable sets are both examples of full trios. An equivalent formulation of the definition of a rational cone is by asking that the family be closed under rational transductions [5, Section V.2].

We note that no mention has yet been made of those operations contributing to the definition of regular expressions. If a family of languages is a rational cone and is also closed under union, we call the family a *semi-AFL*.

If a family of languages is a trio but further is closed under union, concatenation and *positive closure* we say that it is an AFL. The positive closure or $+$-*closure* of a language L is the set

$$L^+ = \bigcup_{i=1}^{\infty} L^i,$$

that is, the set of languages is closed under subsemigroup generation. If an AFL is also closed under arbitrary homomorphism (it is a full trio) we say that it is a *full AFL*.

We may also define a family of languages in terms of one key language. If for some language L the language family $\mathcal{F}$ is the *least* AFL containing L, we say that $\mathcal{F}$ is *principal*. It is also usual to say that the principal AFL is *generated* by L. We summarize this section in Figure 2.1. In Figure 2.2 we compare the closure properties of the language classes which we have seen in this chapter. We denote the regular

languages by REG, the context free languages by CFL, the context sensitive languages by CSL and the recursively enumerable sets by RE.

Included in the table are a number of language families which will be defined in the next chapter. The blind counter languages are denoted by BLIND and the partially blind counter languages by PBLIND. The prefix 1- denotes a single counter, so for example 1-PBLIND denotes the partially blind one counter languages.

The Chomsky hierarchy traditionally refers to the relative inclusions of the classes of regular, context-free, context sensitive and recursively enumerable languages, although some authors now use the term more widely. The four classical language classes are arranged as follows in the hierarchy

$$REG \subset CFL \subset CSL \subset RE.$$

	ϵ-free morphisms	inverse morphisms	arbitrary morphisms	$\cap$ REG	union	concat	+-closure
trio	✓	✓	✓				
faithful cone	✓	✓	✓				
full trio	✓	✓	✓	✓			
rational cone	✓	✓	✓	✓			
semi-AFL	✓	✓		✓	✓		
full semi-AFL	✓	✓	✓	✓	✓		
AFL	✓	✓	✓	✓		✓	✓
full AFL	✓	✓	✓	✓	✓	✓	✓

Table 2.1: The closure properties of various classes of language families.

	semi-AFL	full semi-AFL	AFL	full AFL
REG				✓
CFL				✓
CSL			✓	
RE				✓
1-PBLIND		✓		
PBLIND	✓			
1-BLIND		✓		
BLIND	✓			

Table 2.2: Familiar language families and their closure properties.

Chapter 3

M-automata

In this chapter we introduce the usual definition of a monoid automaton, and consider related results. Many results featuring M-automata for M a specific type of group or monoid are scattered across the computer science literature. Such results have provided important insights in combinatorial group theory and formal language theory. One aim of this chapter is to collect some such results together in a coherent and consistent form. We also establish some new foundational results.

Let M be a monoid with identity 1 and let Σ be a finite alphabet. An *M-automaton* (or *monoid automaton* when we do not need to refer to a specific monoid) over Σ is a finite automaton over the direct product $M \times \Sigma$. We say that the automaton accepts a word $w \in \Sigma^*$ if it accepts $(1, w)$, that is if there exists a path connecting the initial state to some terminal state labelled by $(1, w)$. Intuitively, we visualise an M-automaton as a finite automaton augmented with a memory register which can store an element of M; the register is initialized to the identity element, is modified by right multiplication by element of M, and for a word to be accepted the element present in the memory register on completion must be the identity element. We write $F_1(M)$ for the class of all languages accepted by M-automata, or equivalently for the class of languages accepted by regular M-valence automata [20], that is, finite state automata where each transition is assigned a valence taken from the monoid M. Valence automata are the natural automata theoretic partner to valence grammars - instead of assigning valences to productions, they are assigned to transitions in the

automaton.

We first note a number of well known and obvious results about M-automata languages, which are never the less very useful.

Proposition 3.0.1. *Let $L \subseteq \Sigma^*$ be a language and let $\sigma : X^* \to M$ be a finite choice of distinct generators for a monoid M. Then L is accepted by an M-automaton if and only if it is accepted by an M-automaton having edge labels from M only of the form $m = x\sigma$ where $x \in X \cup \{\epsilon\}$.*

Proof. Let A be an M-automaton accepting the language L, without redundant states and edges. We may write A as an M-automaton with edge labels from M only of the form $x\sigma$ for some $x \in X$ by splitting any edges labelled by $m \in M$ with $m \neq x\sigma$ for some $x \in X$. That is, if $m = (x_1 \dots x_n)\sigma$ (with $x_1, \dots, x_n \in X$) we replace the edge labelled by $(m, w) \in M \times \Sigma^*$ with a sequence of edges, beginning with an edge $(x_1\sigma, w)$ and followed sequentially by edges $(x_i\sigma, \epsilon)$ for $i = 2, \dots, n$. In this way we achieve an automaton with the required condition which accepts the same language as the original M-automaton A. □

Proposition 3.0.2 ([35])**.** *Let $L \subseteq \Sigma^*$ be a language and M be a finitely generated monoid. Then the following are equivalent.*

(i) *L is accepted by an M-automaton;*

(ii) *L is a rational transduction of the identity language of M with respect to some finite generating set;*

(iii) *L is a rational transduction of the identity language of M with respect to every finite generating set.*

Proof. Assume first that (i) is true. We shall prove that (i) implies (iii). Let $\sigma : X^* \to M$ be a finite choice of generators for M. Then by Proposition 3.0.1 there exists an M-automaton A with edge labels from M of the form $m = x\sigma$ where $x \in X \cup \{\epsilon\}$. We construct a rational transducer from X^* to Σ^* from the resulting automaton by replacing edge labels of the form $(x\sigma, w) \in M \times \Sigma^*$ with $(x, w) \in X \times \Sigma^*$. Now $w \in L$

if and only if A has a path from the initial state to some terminal state labelled by $((x_1\sigma)(x_2\sigma)\dots(x_n\sigma), w)$ for some $x_1, \dots, x_n \in X$ such that $(x_1 \dots x_n)\sigma = 1$. But this is true exactly if the transducer has an accepting path labelled $(x_1 \dots x_n, w)$ for some $x_1 \dots x_n$ in the identity language of M. Then, since our choice of generators was arbitrary, (iii) holds.

To show that (ii) implies (i), assume that (ii) holds. Then there exists a finite choice of generators $\sigma : X^* \to M$ and a rational transducer A from X^* to Σ^* such that L is the image of the identity language of M under the transduction. We construct an M-automaton accepting L by replacing each edge label of the form $(x, w) \in X^* \times \Sigma^*$ with $(m, w) \in M \times \Sigma^*$ where $x\sigma = m$. It follows easily that the resulting automaton is an M-automaton accepting the language L.

Since the monoid M is assumed to be finitely generated, it is immediate that (iii) implies (ii), which completes the proof.

□

Another proposition which will be useful is the following.

Proposition 3.0.3. *Let M and N be monoids with N a submonoid of M. Then $F_1(N) \subseteq F_1(M)$.*

Proof. Let A be an N-automaton accepting the language $L \subseteq \Sigma^*$. Since $N \subseteq M$ every edge label in A lies in $M \times \Sigma^*$, so we may regard A as an M-automaton. It is clear from the definitions that it accepts the same language.

□

Before moving on to finite groups, we make some more general observations about finite monoids.

Proposition 3.0.4. *Let M be a monoid. Then $F_1(M)$ contains the regular languages.*

Proof. Let $L \subseteq \Sigma^*$ be a regular language. Then there exists a finite automaton over Σ^* accepting precisely L. Applying the transformation

$$\Sigma^* \to M \times \Sigma^* \quad x \mapsto (1, x)$$

to the edge labels we obtain an M-automaton A accepting precisely the language L as required. □

For the next propositions we require the use of one of Green's relations - recall that two elements $a, b \in S$ are $\mathscr{R}$-related, $a\mathscr{R}b$ if and only if $aS^1 = bS^1$. In the following propositions we will use the fact that for two elements $a, b \in S$, $a\mathscr{R}b$ if and only if there exist elements $s, s' \in S^1$ such that $as = b$ and $bs' = a$. A similar equivalence exists for $\mathscr{L}$-related elements. Recall that for an element $a \in S$, $\mathscr{R}_a$ denotes the $\mathscr{R}$-class containing the element a.

Proposition 3.0.5. *Let M be a finitely generated monoid with $\mathscr{R}_1$ finite. Then $F_1(M)$ is equal to the regular languages.*

Proof. Proposition 3.0.4 above tells us that $F_1(M)$ contains the regular languages, so we need only show that every language in $F_1(M)$ is regular.

Let $\varphi : X^* \to M$ be a finite choice of generators for M and let $L \in F_1(M)$. Then L is a rational transduction of the identity language of M by Proposition 3.0.2. By Theorem 2.2.9 it suffices to show that the identity language of M is regular.

We define a finite automaton over the free monoid X^* with state set $\mathscr{R}_1$ and unique initial and terminal state the identity element. Two states p and q are connected by an edge labelled by $x \in X$ if and only if $p(x\varphi) = q$. Since $\mathscr{R}_1$ is finite and X is finite the state set and edge set of our automaton must be finite, and the automaton accepts precisely the identity language of M. Therefore the identity language of M is a regular language, and the result follows. □

A group G is called *locally finite* if all finitely generated subgroups of G are finite. Mitrana and Stiebe proved the following.

Theorem 3.0.6 ([40])**.** *For any group G, $F_1(G)$ is equal to the regular languages if and only if G is locally finite.*

If we consider locally finite monoids (where all finitely generated submonoids are finite) however, we cannot conclude the same result. Below we shall give an

exact characterisation of monoids M such that $F_1(M)$ is equal to the class of regular languages.

3.1 Cyclic and abelian groups

Since finite groups have been covered implicitly in the previous section, we next consider the case of cyclic groups. We need only consider the infinite cyclic group $\mathbb{Z} = \langle x \rangle$. $\mathbb{Z}$-automata are sometimes also referred to as *blind one-counter automata*, where they are presented as finite automata augmented with a single integer counter which cannot be read. We will use both notations interchangeably.

For a group G and a property P (for example, the property of being finite, cyclic, abelian, free) we say that the group G is *virtually* P if there exists a subgroup of finite index in G which has the property P. From the perspective of word problems of cyclic groups, a result of Herbst [31], extended by Elston and Ostheimer [18] is the following.

Theorem 3.1.1. *Let G be a finitely generated group. Then the word problem of G is accepted by a $\mathbb{Z}$-automaton if and only if G is virtually cyclic.*

The natural next class of groups to consider are the free abelian groups of rank n. A free abelian group of rank n is isomorphic to $\mathbb{Z}^n$, a direct product of n cyclic groups. Again, it is straight forward to see that the definition of $\mathbb{Z}^n$-automata is equivalent to that of *blind n-counter machines* [35]. The proof of the corresponding result about word problems is much more involved than for cyclic groups however.

Theorem 3.1.2 ([16])**.** *Let G be a finitely generated group. The word problem of G is accepted by a $\mathbb{Z}^n$-automaton if and only if G is virtually free abelian of rank n or less.*

The result is proved by establishing bounding results for minimal elements of intersections of semilinear sets. These results are then applied to conclude that a group whose word problem is accepted by a $\mathbb{Z}^n$-automaton must have polynomial growth of degree less than n. A seminal result of Gromov [28] states that a group has

polynomial growth if and only if the group is virtually nilpotent, and thus G must be virtually nilpotent in the case of the theorem. Finally applying a combinatorial result of Mitrana and Stiebe [40], the result is achieved.

3.2 Free groups

Recall the formal categorical definition of a free group from Chapter 2. The (unique up to isomorphism) free group on n generators has monoid presentation

$$F_n = \langle x_1, \ldots, x_n, x_1^{-1}, \ldots, x_n^{-1} \mid x_1 x_1^{-1} = x_1^{-1} x_1 = \ldots = x_n x_n^{-1} = x_n^{-1} x_n = 1 \rangle.$$

The free groups provide the basis for all study in combinatorial group theory since any group G is a quotient of a free group. Indeed, if G is a group there exists a free group F and a normal subgroup N of F such that $G \cong F/N$, that is, G is isomorphic to the quotient of F by N.

An important property of free groups is the following.

Theorem 3.2.1 ([38]). *The free group on n letters for $n \geq 2$ embeds in the free group on two letters, F_2.*

This result often allows us to talk just about the free group on two letters.

The Dyck languages consist of balanced strings of parenthesis, so the word (()()) would be included but the word ()((would not. The *one-sided Dyck language* allows only pairing of parentheses in the usual way, so we may pair and cancel () but not)(. When both of these pairings are allowed we call the resulting language the *two-sided Dyck language*. Chomsky and Schützenberger made the following important observation.

Theorem 3.2.2 ([9]). *Let $L \subseteq \Sigma^*$ be a language. The following are equivalent.*

(i) *L is context-free.*

(ii) *L is a rational transduction of the one-sided Dyck language on two pairs of parentheses.*

(iii) *L is a rational transduction of the two-sided Dyck language on two pairs of parentheses.*

Even without an understanding of rational transductions, it is easy to see the equivalence between the two-sided Dyck language and the word problem of the free group. For example, consider the two-sided Dyck language on two pairs of parentheses and the free group of rank two generated by x_1 and x_2. Applying a straight forward substitution:

$$(\rightarrow x_1, \quad) \rightarrow x_1^{-1},$$
$$[\rightarrow x_2, \quad] \rightarrow x_2^{-1},$$

we can see the equivalence of the word

$$(()[][])$$

from the two-sided Dyck language with the word

$$x_1 x_1 x_1^{-1} x_2 x_2^{-1} x_2 x_2^{-1} x_1^{-1} = 1$$

from the free group. So using Proposition 3.0.2 we may restate the equivalence of parts (i) and (iii) of Theorem 3.2.2 as follows.

Theorem 3.2.3. *Let $L \subseteq \Sigma^*$ be a language. Then L is context-free if and only if L is accepted by a F_2-automaton.*

A direct algebraic proof of this result was first claimed by Mitrana and Dassow [13], however the proof was incorrect as described in [11]. A correct proof was provided by Corson [11]. The observation of equivalence between this result and part of the Chomsky and Schützenberger result was made in [35].

We will deal with the other the equivalence of parts (i) and (ii) of the Chomsky and Schützenberger theorem in the following section.

Results of Muller and Schupp [42, 43] combined with a result of Dunwoody [14] give a result about word problems for the context-free case.

Theorem 3.2.4. *Let G be a finitely generated group. The word problem of G is context-free if and only if G is virtually free.*

Letting $X = \{x_1, x_2, x_1^{-1}, x_2^{-1}\}$ and $Y = \{y_1, y_2, y_1^{-1}, y_2^{-1}\}$ be two disjoint sets we define the group $F_2 \times F_2$ with monoid presentation

$$\langle X, Y \mid x_1x_1^{-1} = x_1^{-1}x_1 = x_2x_2^{-1} = x_2^{-1}x_2 = 1 \; xy = yx, x \in X, y \in Y\rangle,$$

the direct product of two copies of the free group on two letters. With respect to this group, Mitrana and Stiebe observed another interesting property.

Theorem 3.2.5 ([41])**.** *$F_1(F_2 \times F_2)$ is exactly the family of recursively enumerable languages.*

3.3 Polycyclic monoids

Let X be a set. The *polycyclic monoid* on X is the monoid $P(X)$ generated, under the operation of composition of relations, by the partial bijections of the form

$$p_x : X^* \to X^*, \quad w \mapsto wx$$

and

$$q_x : X^*x \to X^*, \quad wx \mapsto w.$$

The monoid $P(X)$ is a natural algebraic model of a pushdown store or stack on the alphabet X, with p_x and q_x corresponding to the elementary operations of pushing x and popping x (where defined) respectively, and composition to performing these operations in sequence.

Clearly for any $x \in X$, the composition p_xq_x is the identity map. On the other hand, if x and y are distinct letters in X, then p_xq_y is the *empty map* which constitutes a zero element in $P(X)$. In the case $|X| = 1$, say $X = \{x\}$, the monoid $P(X)$ is called the *bicyclic monoid*, and is often denoted B. The partial bijections p_x and q_x alone (which we shall often denote just p and q) do not generate the empty map, and so the bicyclic monoid does not have a zero element; to avoid having to treat it as a

special case, it is convenient to write $P^0(X)$ for the union of $P(X)$ with the empty map; thus we have $P^0(X) = P(X)$ if $|X| \geq 2$ but $P^0(X)$ isomorphic to $P(X)$ with a zero adjoined if $|X| = 1$.

Let $P_X = \{p_x \mid x \in X\}$ and $Q_X = \{q_x \mid x \in X\}$, and let z be a new symbol not in $P_X \cup Q_X$ which will represent the zero element. Let $\Sigma_X = P_X \cup Q_X \cup \{z\}$. Then there is an obvious surjective morphism $\sigma : \Sigma_X^* \to P^0(X)$, and indeed $P^0(X)$ admits the monoid presentation

$$P^0(X) = \langle \Sigma_X \mid p_x q_x = 1, p_x q_y = z,$$
$$zp_x = zq_x = p_x z = q_x z = zz = z \text{ for all } x, y \in X,\ x \neq y \rangle.$$

Returning to the Chomsky and Schützenberger result for context-free languages (Theorem 3.2.2), we conclude that the identity language of $P(X)$ ($|X| = n, n \geq 2$) is precisely equivalent to the one-sided Dyck language on $2n$ letters.

Theorem 3.3.1 ([22, 34])**.** *For $|X| \geq 2$ a $P(X)$-automaton is equivalent to a push-down automaton with stack alphabet X, so that the language class $F_1(P(X))$ is exactly the class of context-free languages.*

The bicyclic monoid and counter automata

The *bicyclic monoid* is the simplest example of a polycyclic monoid, though from a language theoretic perspective it is arguably the most interesting. It has presentation

$$\langle p, q \mid pq = 1 \rangle$$

but can be thought of more easily as being the monoid of operations on a counter which cannot take negative values. Let p denote 'add one to the counter' and let q denote 'subtract one from the counter'. Then if we read the string pq the net effect is the identity. Note that $qp \neq 1$, since this would go against our assumption that we cannot drop below zero in our counter. B-automata are precisely *partially blind one-counter automata* as defined by Greibach [26], and hence B^n-automata are also referred to as *partially blind n-counter automata*. As with their blind counterparts,

we will use both notations interchangeably. We note that the identity language of B is precisely the one-sided Dyck language on a single pair of parentheses. Elements of the bicyclic monoid then take the form $q^m p^n$ with m, n non-negative integers.

We have already noted the equivalence of the one-sided Dyck language on n pairs of parenthesis to the identity language of the polycyclic monoid of order n. Hence the identity language of the bicyclic monoid is equal to the one-sided Dyck language on a single pair of parenthesis. Similarly we have observed the equivalence of the two-sided Dyck language on n pairs of parenthesis to the word problem of the free group on n letters when $n \geq 2$. It is easy to see that for a single pair of parenthesis we have precisely the word problem of $\mathbb{Z}$.

Proposition 3.3.2 ([6])**.** *The one-sided Dyck language on one pair of parenthesis is not the image of the two-sided Dyck language on one pair of parenthesis under a rational transduction, and vice versa.*

Combining this with Proposition 3.0.2 we may conclude:

Theorem 3.3.3. *$F_1(\mathbb{Z})$ and $F_1(B)$ are incomparable under inclusion.*

While $F_1(B)$ clearly contains only context-free languages, if we consider M-automata defined over the direct product of two copies of the bicyclic monoid, that is, partially blind two-counter automata, we have the following result.

Proposition 3.3.4. *$F_1(B^2)$ is not contained in the set of context-free languages.*

Proof. We claim that $F_1(B^2)$ contains languages such as

$$L = \{a^i b^j c^i d^j \mid i, j \geq 1\}$$

which do not satisfy the pumping lemma for context-free languages. Indeed, let B_1 and B_2 be two disjoint copies of the bicyclic monoid, with sets of generators $\{p_1, q_1\}$ and $\{p_2, q_2\}$ respectively. Then the language above is accepted by the $B_1 \times B_2$-automaton shown in Figure 3.1.

It suffices to show that the language L does not satisfy the pumping lemma for context-free languages (Lemma 2.3.1). We assume for a contradiction that L

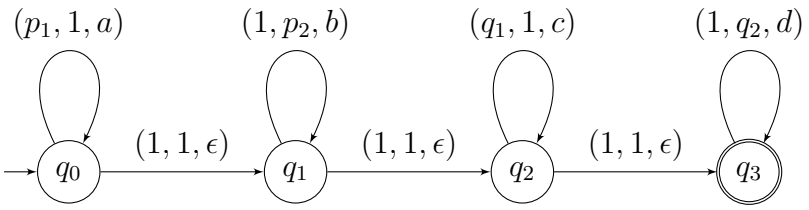

Figure 3.1: A B^2-automaton accepting the language $\{a^i b^j c^i d^j \mid i, j \in \mathbb{N}\}$.

satisfies the lemma, and let $m \in \mathbb{N}$ be the pumping length for L. Consider a word $z = a^m b^n c^m d^n \in L$ with $n > m$. Clearly $|z| \geq m$. Now we must consider where the strings to be pumped must lie within the word z. Recall that to satisfy the pumping lemma, we must first be able to factorise the word as $z = uvwxy$ so that $uv^j wx^j y \in L$ for all $i \geq 1$ and $|vxy| \leq m$. Clearly we have two options. Either

(i) We let $v = a^i$ and $y = c^i$ for some $1 \leq i < m$ or

(ii) We let $v = b^i$ and $y = d^i$ for some $1 \leq i < m$.

Recall that a condition of the pumping lemma states that the subword vxy must have length less than or equal to the pumping length m. But in case (i) the length of vxy is at least n which was defined to be greater than m. In case (ii), the length of vxy must also be strictly greater than m, and hence we cannot satisfy the conditions of the pumping lemma and we have a contradiction.

Therefore the language L does not satisfy the pumping lemma for context-free languages, and so by Proposition 3.0.2 there exists a language in $F_1(B^2)$ which is not context-free. □

3.4 Nilpotent groups

Let H and K be normal subgroups of a group G. If H/K is contained in the centre of G/K then H/K is called a *central factor* of G. A group G is *nilpotent* if and only if it has a finite series of normal subgroups

$$G = G_0 \geq G_1 \geq \ldots \geq G_r = 1$$

such that G_{i-1}/G_i is a central factor of G for each $i = 1, \ldots, r$. The smallest value of the length r of such a series for a group G is called the *nilpotency class* of G. So for example, abelian groups are nilpotent of class 1.

Results relating G-automata and word problems have so far been limited for G a nilpotent group. However automata over nilpotent groups accept a class of languages which have some interesting properties, and for this reason we briefly mention them. One result is the following.

Theorem 3.4.1 ([21])**.** *Let G be a finitely generated nilpotent group of class c. Then the word problem of G is context sensitive.*

It has been claimed in [15] that this result combined with Proposition 3.0.2 is sufficient to imply a result similar to Theorems 3.0.6 and 3.2.3 above for nilpotent groups: a result saying that languages accepted by G-automata where G is a nilpotent group are context-sensitive. However, this would require the closure of the context-sensitive languages under rational transductions. The family of context-sensitive languages is not closed under arbitrary morphisms and hence since arbitrary morphisms are examples of rational transductions, the family of context sensitive languages is not closed under rational transductions [39]. Thus all we may really conclude is that the G-automata languages where G is a finitely generated nilpotent group of class c are recursively enumerable, a significantly weaker result.

The discrete Heisenberg group

In this section we explore the formal language properties of one of the simplest of the nilpotent groups, the discrete Heisenberg group. Recall that a group G is called *torsion* if every element has finite order, that is, for each $x \in G$ there exists some $n \in \mathbb{N}$ such that $x^n = 1$, the identity element of the group. A group is *torsion-free* if the only element of finite order is the identity element.

The discrete Heisenberg group is a non-abelian, torsion-free, nilpotent group of class two. It is one of the simplest examples of a nilpotent group to present and understand. It may be presented as a matrix group generated by the following two

3×3 matrices.

$$a = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

The form of the generators imply the relations

$$[a, [a, b]] = 1, \quad [b, [a, b]] = 1,$$

(where in this case square brackets denote the *commutator* $a^{-1}b^{-1}ab$). In fact these relations suffice to define the group, so that it has presentation

$$\langle\ a, b \mid [a, [a, b]] = 1 = [b, [a, b]]\ \rangle.$$

Thinking in terms of a group presentation it is more straightforward to define a new generator $c = [a, b]$ giving the presentation

$$\langle a, b, c \mid ab = bac, ac = ca, bc = cb \rangle.$$

The central series of H has the form

$$\{1\} \leq \langle c \rangle \leq H$$

where $\langle c \rangle = [H, H] = Z(H)$ (where $[H, H]$ denotes the *commutator subgroup*, the subgroup generated by all the commutators).

Since H is torsion-free and finitely generated we may refine the upper central series to form a central series

$$H = H_0 > H_1 > \ldots > H_n = 1$$

for which each H_{i-1}/H_i is an infinite cyclic group. Then for H we have the following:

$$H > \langle b, c \rangle > \langle c \rangle > 1.$$

We now choose elements u_i to form our *canonical basis* [30] such that G_{i-1} is generated by G_i and u_i for each $i = 0, \ldots n$ where in our case $n = 2$. This allows us to write any element $x \in H$ in the form $x = u_1^i u_2^j u_3^k$ where $u = (u_1, u_2, u_3)$ is the canonical basis

and for some $i, j, k \in \mathbb{Z}$ the *canonical parameters* of x. Hence the canonical basis of H is $u = (a, b, c)$ so that any element $x \in H$ may be written uniquely as $x = a^i b^j c^k$.

We note the close association of the Heisenberg group with the naive quadratic sorting algorithm *bubble sort*. Starting with a word consisting of letters a, b and their inverses, we convert the word to normal form, by commuting a's and b's, thus adding powers of c to the end of the word. In fact, the result is an alphabetized word, with the power of c encoding the number of 'swaps' which were necessary.

A useful way of presenting elements of the Heisenberg group is as integer triples representing the canonical parameters of a given element in H. Then the standard matrix multiplication in the group appears very differently and we have the following.

$$(a^i b^j c^k) \cdot (a^u b^v c^w) = (a^{i+u}, b^{j+v}, c^{k+w+uj}).$$

In terms of automata in this presentation, we can look at incrementing the 'counters' as follows. In fact what we are able to do is view a, b and c as operators on the group (by right multiplication) and hence on the three counters.

$$\begin{aligned}(i, j, k) \cdot (1, 0, 0) &= (i + 1, j, k);\\ (i, j, k) \cdot (0, 1, 0) &= (i, j + 1, k + i);\\ (i, j, k) \cdot (0, 0, 1) &= (i, j, k + 1).\end{aligned}$$

In order to demonstrate the interesting properties present in languages accepted by nilpotent group automata, we include three examples of *Heisenberg automata*, that is, G-automata where G is the discrete Heisenberg group, whose languages demonstrate some form of multiplication. In Figure 3.2 we have a Heisenberg automaton accepting the language $\{x^p y^q z^{pq} \mid p, q \geq 0\}$. Indeed, an accepting path through the automaton must have label $(a^p b^q a^{-p'} b^{-q'} c^{-r}, x^p y^q z^r)$ for some $p, q, r \in \mathbb{N}$. Using the normal form for H we may conclude:

$$\begin{aligned}(a^p b^q a^{-p'} b^{-q'} c^{-r}, x^p y^q z^r) &= (x^p y^q z^r, a^p a^{-p'} b^q b^{-q'} c^{-r})\\ &= (x^p y^q z^r, a^{p-p'} b^{q-q'} c^{p'q-r}).\end{aligned}$$

Then $a^{p-p'}b^{q-q'}c^{p'q-r} = 1$ in H if and only if $p = p'$, $q = q'$ and hence $p'q = pq = r$ and the automaton accepts precisely $\{x^p y^q z^{pq} \mid p, q \geq 0\}$ as required.

In Figure 3.3 we have a Heisenberg automaton accepting composite numbers. An accepting path through the automaton must have label $(a^p b^q a^{-p'} b^{-q'} c^n, x^n)$ and using the normal form as above we see

$$(a^p b^q a^{-p'} b^{-q'} c^n, x^n) \;=\; (a^{p-p'} b^{q-q'} c^n, x^n)$$

and since $a^{p-p'}b^{q-q'}c^n = 1$ in H we have $p = p'$ and $q = q'$, so $n = pq$ and the automaton accepts precisely the set $\{x^{pq} \mid p, q > 1\}$ as required.

In Figure 3.4 we have a Heisenberg automaton accepting the language $\{x^p y^{pn} \mid p \in \mathbb{N}\}$. An accepting path through the automaton has label $((ab^n a^{-1} b^{-n})^p c^{-p'}, x^p y^{p'})$. Reasoning as before, we use the normal form to conclude

$$\begin{aligned} ((ab^n a^{-1} b^{-n})^p c^{-p'}, x^p y^{p'}) &= ((a^{1-1} b^{n-n})^p c^{pn-p'}, x^p y^{p'}) \\ &= (c^{pn-p'}, x^p y^{p'}). \end{aligned}$$

The path is accepting if and only if $c^{pn-p'} = 1$ in H and so $p' = pn$ as required.

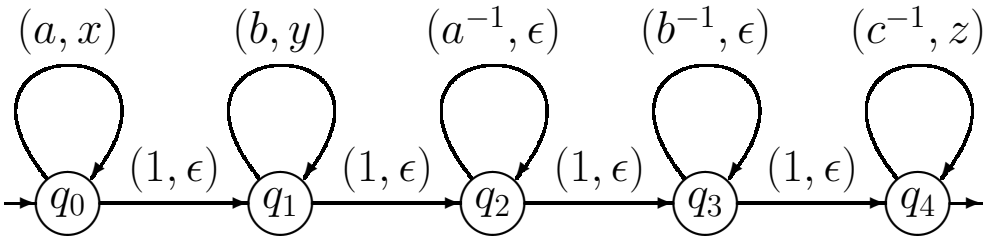

Figure 3.2: A H-automaton accepting the set $\{x^p y^q z^{pq} \mid p, q \geq 0\}$

With respect to the existing language classes covered in this thesis, we make the following obvious observation.

Proposition 3.4.2. *$F_1(\mathbb{Z}^2) \subseteq F_1(H)$.*

Proof. Since $H \geq \langle a, c \rangle \cong \mathbb{Z}^2$, we see that $\mathbb{Z}^2$ is a submonoid of H. Then by Proposition 3.0.3 the result follows. □

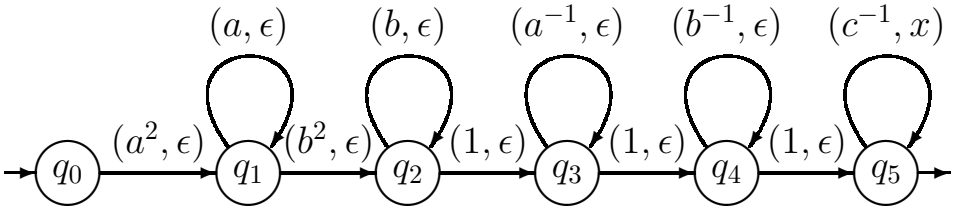

Figure 3.3: A H-automaton accepting the set $\{x^{pq} \mid p, q > 1\}$

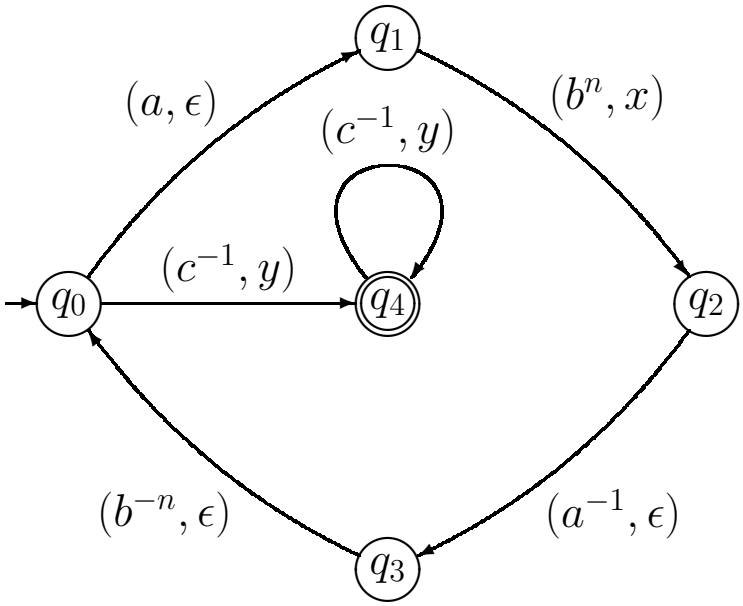

Figure 3.4: A H-automaton accepting the set $\{x^p y^{pn} \mid p \in \mathbb{N}\}$

Consequently we also have $F_1(\mathbb{Z}), REG \subseteq F_1(H)$. So we conclude that even for a relatively simple choice of nilpotent group, the positioning of the corresponding language class within the Chomsky hierarchy is already very interesting. This subject is deserving of further study.

Chapter 4

Monoid automata and their extensions

In this chapter we consider the properties of M-automata over general monoids and semigroups, and what effect extending the definition of monoid automata has on these properties. We first examine the interactions of M-automata with the structure of a given monoid, noting some limitations on the power of M-automata which result. We then consider a natural extension to the definition which circumvents some of these limitations. Some of the material in this chapter has been published in [49, 50, 51].

4.1 The structure of a monoid

The aim of this section is to show that the extent to which an M-automaton can make use of the structure of a general monoid M is severely limited. There are many interesting structural properties of monoids, such as ideals, identity and zero elements, which as we shall see in what follows, can effect the way in which monoid automata behave. Finally in this section we use our observations to present a theorem outlining the types of language class which can be derived from monoid automata.

Proposition 4.1.1. *Let I be a proper ideal of a monoid M. Then $F_1(M) = F_1(M/I)$.*

Proof. Suppose $L \in F_1(M)$, and let A be an M-automaton accepting L. First notice that any path containing an edge of the form (x, w) with $x \in I$ will itself have label

with first component in I; in particular, since I is a proper ideal, $1 \notin I$ and such a path cannot be an accepting path. It follows that we may remove any such edges without changing the language accepted, so that we may assume without loss of generality that A has no such edges. Now for any $x_1, \ldots, x_n \in M \setminus I$, it follows from the definition of M/I that $x_1 \ldots x_n = 1$ in M if and only if $\{x_1\} \ldots \{x_n\} = \{1\}$ in M/I. If we let B be the (M/I)-automaton obtained from A by replacing edge labels of the form (x, w) with $(\{x\}, w)$, it follows from the above fact that A has a path from the initial vertex to a terminal vertex labelled $(1, w)$ if and only if B has a path from the initial vertex to a terminal vertex labelled $(\{1\}, w)$. Hence B accepts the language L and $L \in F_1(M/I)$.

Conversely, if $L \in F_1(M/I)$ then L is accepted by some (M/I)-automaton. We may assume without loss of generality that B has no edges labelled by the zero element I. Indeed let $w \in L$ and assume that there exists an edge in the accepting path labelled by w which is labelled by the zero element I. Then the cumulative label of the whole path must be I, which contradicts our assumption. We now obtain from B a new M-automaton A by replacing edge labels of the form $(\{x\}, w)$ with (x, w). Since for $x_1, \ldots, x_n \in M \setminus I$, $x_1 \ldots x_n = 1$ if and only if $\{x_1\} \ldots \{x_n\} = \{1\}$, any accepting path through B will also be an accepting path in A. So A accepts exactly L, and so $L \in F_1(M)$.

□

Recall that a monoid M is called *simple* if it does not contain any proper ideals. Similarly a monoid M with zero is called *0-simple* if the only ideals are $\{0\}$ and M itself and additionally $M^2 \neq \{0\}$. The latter condition excludes only the 2 element null semigroup and forces $M^2 = M$.

Corollary 4.1.2. *For every monoid M there is a simple or 0-simple monoid N such that $F_1(M) = F_1(N)$.*

Proof. If M has no proper ideals then it is simple, so we are done. Otherwise, let I be the union of all the proper ideals of M. Then I is an ideal and, since the identity element 1 does not lie in any proper ideal, $1 \notin I$ and I is a proper ideal of M. Set

$N = M/I$ and assume for a contradiction that there exists some $J \subseteq N$, a proper non-zero ideal. But if J is an ideal of N, $J' = \{x \in M \mid \{x\} \in J\} \cup I$ must be a proper ideal of M. But this contradicts our assumption that N was exactly the result of removing all proper ideals from M, and so $J = N$ and N has no proper non-zero ideals. Hence either $N^2 = \{0\}$ or N is 0-simple. In the former case N is the 2 element null semigroup so by Proposition 3.0.5, $F_1(N) = REG = F_1(\{1\})$ where $\{1\}$ is the trivial monoid which is simple. Otherwise N is 0-simple and by Proposition 4.1.1 we have $F_1(M) = F_1(M/I) = F_1(N)$ as required. □

Corollary 4.1.2 tells us that the usual theory of M-automata really only involves the very restricted classes of simple and 0-simple monoids. The following proposition deals with a special case with respect to zero, and says that we may restrict our study further in this particular situation.

Proposition 4.1.3. *Let M be a monoid. Then $F_1(M^0) = F_1(M)$.*

Proof. That $F_1(M) \subseteq F_1(M^0)$ follows immediately from Proposition 3.0.3 since $M \subset M^0$, so we need only prove the converse. Suppose $L \in F_1(M^0)$, and let A be an M^0-automaton accepting L.

As in the proof of Proposition 4.1.1 we first note that any path containing an edge labelled by zero must itself have label zero. However, for a word to be accepted we must have the identity element present in the memory register on reaching a terminal state in the automaton, and hence we conclude that any accepting path through A must not contain an edge with first component zero.

It follows that we may remove all edges whose first label component is zero from the automaton without affecting the language accepted, obtaining a new M^0-automaton B which accepts the language L. But now since M is a submonoid of M^0, B may be interpreted as an M-automaton accepting L, so that $L \in F_1(M)$ as required. □

Recall that an *idempotent* element e in a semigroup S is an element such that $ee = e$. Further, an idempotent element e is called *primitive* if for every non-zero

idempotent f such that $ef = fe = f$ we have $e = f$. A semigroup is *completely simple* [respectively, completely 0-simple] if it is simple [0-simple] and has a primitive idempotent. For more information about completely simple and completely 0-simple semigroups, see [33]. An alternative but very useful characterisation of completely simple and completely 0-simple semigroups comes from the Rees theorem, which we outline below.

Let T be a semigroup, 0 be a new symbol not in T and let I, J be non-empty sets. Let $P = (P_{ji})$ be a $J \times I$ matrix with entries in $T \cup \{0\}$. We define a new semigroup with set of elements

$$(I \times T \times J) \ \cup \ \{0\}$$

and multiplication defined by

$$(i,t,j)(i',t',j') = \begin{cases} (i, tP_{ji'}t', j') & \text{if} P_{ji'} \neq 0 \\ 0 & \text{otherwise,} \end{cases}$$

and

$$(i,t,j)0 = 0(i,t,j) = 00 = 0.$$

It is simple to verify that this binary operation is associative; we call the semigroup constructed in this way a *Rees matrix semigroup with zero over* T, and denote it $M^0(T; I, J; P)$. The semigroup T is called the *base semigroup* and the matrix P the *sandwich matrix* of the construction. If P contains no zero entries then $I \times T \times J$ forms a subsemigroup of $M^0(T; I, J; P)$, called a *Rees matrix semigroup (without zero) over* T and denoted $M(T; I, J; P)$.

Rees matrix semigroups play a crucial role in much of the structural theory of semigroups. Of particular importance is the case that the base semigroup T is a group G. A Rees matrix semigroup with zero over a group is called *regular* [10] if every row and every column of the sandwich matrix contains a non-zero entry. The importance of this construction can be seen from the following seminal result of Rees [48].

Theorem 4.1.4 (The Rees Theorem)**.** *Let* $S = M^0(G; I, J; P)$ *be a regular Rees*

matrix semigroup constructed as above with G a group. Then S is a completely 0-simple semigroup. Conversely, every completely 0-simple semigroup is isomorphic to one constructed in this way.

As a corollary, we have a similar result for semigroups without zero.

Corollary 4.1.5. *Let $S = M(G; I, J; P)$ be a Rees matrix semigroup (without zero) constructed as above with G a group. Then S is completely simple. Conversely, every completely simple semigroup is isomorphic to one constructed in this way.*

The final aim in this section will be to provide a theorem classifying possibilities for $F_1(M)$ for M a monoid. We first note some useful results from [10].

Theorem 4.1.6. *Let e be a non-zero idempotent of a 0-simple semigroup S which is not completely 0-simple. Then S contains a copy of the bicyclic monoid having e as an identity element.*

Proposition 4.1.7. *A completely 0-simple semigroup contains an identity element if and only if it is isomorphic to G^0 for some group G.*

Proof. Let $S = M^0(G; I, J; P)$ be a completely 0-simple semigroup containing an identity element $e = (i, g, j)$ say. For every $i' \in I$ we have

$$(i', g, j) = e(i', g, j) = (i, gP_{ji'}, j)$$

and so we may conclude that $i = i'$. Thus $|I| = 1$ and symmetrically $|J| = 1$. Clearly $P_{ji} \neq 0$ and so $S \setminus \{0\}$ is a subsemigroup. It will therefore suffice to show that $S \setminus \{0\}$ is a group. Let $(i, h, j) \in S \setminus \{0\}$ and consider the element $(i, P_{ji}^{-1}h^{-1}g, j)$. Then

$$(i, h, j)(i, P_{ji}^{-1}h^{-1}g, j) = (i, hP_{ji}P_{ji}^{-1}h^{-1}g, j) = (i, g, j) = e.$$

So every element in $S \setminus \{0\}$ has a right inverse, and so we conclude that $S \setminus \{0\}$ is a group as required. The converse is clear.

□

We use these facts to prove the following, which is well known.

Corollary 4.1.8. *A simple [0-simple] monoid with identity e is either a group [respectively, a group with 0 adjoined] or contains a copy of the bicyclic monoid as a submonoid having e as its identity element.*

Proof. Let M be a 0-simple monoid with identity element 1. If M is completely 0-simple then Proposition 4.1.7 tells us that M is a group with zero adjoined. If M is not completely 0-simple we may apply Theorem 4.1.6 to conclude that M contains a copy of the bicyclic monoid as a submonoid.

Now let M be a simple monoid (that is, M contains no zero element). If M is completely simple then we may adjoin a zero element to give a completely 0-simple monoid M^0. We may then apply Proposition 4.1.7 to see that M must be a group. If M is not completely simple, then after adjoining a zero we may apply Theorem 4.1.6 to conclude that M^0 contains a copy of the bicyclic monoid. Let $N \subseteq M$ be a subsemigroup isomorphic to the bicyclic monoid. Since the bicyclic monoid B does not contain a zero element, $0 \notin N$. So $N \subseteq M^0$ and the simple monoid M contains a copy of the bicyclic monoid as a submonoid.

□

Recall that a group is called *torsion* if every element has finite order, that is, for each $x \in G$ there exists some $n \in \mathbb{N}$ such that $x^n = 1$. Combining the previous proposition with Propositions 4.1.1 and Corollary 4.1.7 we now obtain the following.

Theorem 4.1.9. *Let M be a monoid. Then either $F_1(M) = F_1(G)$ for some group G, or $F_1(M)$ contains the partially blind one-counter languages.*

Proof. Let M be a monoid. Corollary 4.1.2 tells us that $F_1(M)$ is equal to $F_1(N)$ for some simple or 0-simple monoid N. Corollary 4.1.8 says that N is either a group (or a group with zero adjoined) or contains a copy of the bicyclic monoid. If N is a group then we are done. If N is a group with zero adjoined we may apply Proposition 4.1.3 to see that $F_1(N) = F_1(G)$ for some group G. The remaining possibility is that N contains a copy of the bicyclic monoid, in which case $F_1(N)$ contains the partially blind one-counter languages. □

Before proving the key result of this section, we wish to recall Mitrana and Stiebe's result, appearing as Theorem 3.0.6 in this thesis, which says that $F_1(G)$ is equal to the regular languages if and only if G is locally finite. We have already observed that this result does not hold in the monoid case, but we are now in a position to prove the following.

Theorem 4.1.10. *For any monoid M, $F_1(M)$ is equal to the regular languages if and only if every finitely generated submonoid of M has $\mathscr{R}_1$ finite.*

Proof. Let M be a monoid in which every finitely generated submonoid has $\mathscr{R}_1$ finite. Let $L \subseteq F_1(M)$ and let A be an M-automaton accepting L. Let Y be the submonoid of M generated by elements which appear as edge labels in A. Then A is a Y-automaton accepting L, so $L \in F_1(Y)$. Now Y is a finitely generated submonoid of M, so by assumption Y has finite $\mathscr{R}_1$ class. It follows by Proposition 3.0.5 that L is regular.

For the converse of the theorem we prove the contrapositive statement that if every finitely generated submonoid of M does not have finite $\mathscr{R}_1$ class there must exist non-regular languages in $F_1(M)$. Suppose then that M has a finitely generated submonoid N with $\mathscr{R}_1$ infinite. Consider the Rees quotient monoid $N' = N/(N \setminus \mathscr{J}_1)$ where $\mathscr{J}_1$ is the $\mathscr{J}$ class of 1. Of course, $N' \cong N \setminus (N \setminus \mathscr{J}_1) \cup \{0\} = \mathscr{J}_1 \cup \{0\}$, unless $0 \notin N$ in which case $N' = \mathscr{J}_1$. Hence N' consists of a single $\mathscr{J}$-class and we can conclude [10] that N' is 0-simple (or simple in the case that $0 \notin N$). By Proposition 4.1.1 $F_1(N) = F_1(N')$. Our assumption that $\mathscr{R}_1$ is infinite tells us that N' is infinite since $\mathscr{R}_1$ is contained in N'. We may also conclude that N' is finitely generated since N' is a quotient of a finitely generated monoid N.

If N' is completely simple or completely 0-simple then, since it is a monoid and hence contains an identity element, by Proposition 4.1.7 N' is equal to a group G or a group with zero adjoined G^0. Applying Proposition 4.1.3 we see that in either case $F_1(N) = F_1(N') = F_1(G)$. We have already established that N' is finitely generated and hence that G is finitely generated, and that N' is infinite. Hence G is not locally finite and we may apply Theorem 3.0.6 to conclude that $F_1(G)$ must

contain a non-regular language.

If N' is not completely simple or completely 0-simple then by Proposition 4.1.8 N' contains a copy of the bicyclic monoid as a submonoid. Then $F_1(B)$ is contained in $F_1(N')$ and $F_1(N')$ contains non-regular languages. But since $F_1(N') \subseteq F_1(M)$ we conclude that $F_1(M)$ contains non-regular languages. □

Proposition 4.1.11. *Let M be a monoid. Then $F_1(M)$ either*

(i) *is equal to the regular languages;*

(ii) *contains the blind one-counter languages;*

(iii) *contains the partially blind one-counter languages or*

(iv) *is equal to $F_1(G)$ for G an infinite torsion group which is not locally finite.*

Proof. By Theorem 4.1.9 either $F_1(M)$ contains the partially blind one-counter languages, or $F_1(M) = F_1(G)$ for some group G. In the former case it is immediate that (iii) holds. So suppose $F_1(M) = F_1(G)$ for some group G.

If G is not a torsion group then it has an element of infinite order; this element generates a subgroup isomorphic to $\mathbb{Z}$, from which it follows that $F_1(G)$ contains the class $F_1(\mathbb{Z})$ of blind one-counter languages and (ii) holds. Now by [35, Proposition 1], every language in $F_1(G)$ is in $F_1(H)$ for some finitely generated subgroup H of G. If G is locally finite, then such an H must be finite, and so every language in $F_1(G)$ is regular. Since $F_1(G)$ certainly contains the regular languages, (i) holds. There remains only the case in which G is a torsion group which is not locally finite, in which case (iv) holds. □

We next aim to establish some mutual exclusivity properties on the conditions of Proposition 4.1.11. The following observation was made by Elder and Mintz [17], but has not been published. The proof given is the author's own.

Proposition 4.1.12. *Let G be a torsion group, and let H be a group. If the word problem of H is accepted by a G-automaton then H is torsion.*

Proof. We assume for a contradiction that the word problem of H is accepted by a G-automaton A but H is not a torsion group. Then there exists an element $h \in H$ which has infinite order. Let $\sigma : X^* \to H$ be a choice of generators for H; clearly we may choose σ such that there exists $w \in X$ with $w\sigma = h$. Then $ww^{-1} = 1$ and is contained in the word problem of H. For $i \in \mathbb{N}$, we also may conclude that $w^i w^{-i} \in WP(H)$ and hence is accepted by A. Let π_i denote an accepting path for $w^i w^{-i}$ in A. For some i, there must exist a loop in π_i, of the form (g, w^k) for some $k > 0$ and $g \in G$. We call this loop τ. Since G is torsion, there exists some $l > 0$ such that $g^l = 1$. Now by iterating the loop τ $l + 1$ times we obtain a new loop τ^{l+1} labelled $(g^{l+1} = g, w^{lk+k})$.

We now replace τ with the new loop τ^{l+1} in the path π_i. Since τ and τ^{l+1} have the same label from G the new path is accepting, and has label $w^{i+lk}w^{-i}$ from H. Hence the element $w^{i+lk}w^{-i}$ is contained in the word problem of H. But then $h^{lk} = 1$, which contradicts our assumption that h had infinite order. Therefore we conclude that any group H whose word problem is accepted by a G-automaton must be torsion.

□

And so with respect to Proposition 4.1.11 above we have:

Corollary 4.1.13. *Let G be an infinite torsion group. Then $F_1(G)$ cannot contain the blind one-counter languages.*

Proof. In Section 3.1 we noted that the blind one-counter languages are defined to be precisely the languages accepted by $\mathbb{Z}$-automata. In particular, the word problem of $\mathbb{Z}$ is a blind one-counter language. Since $\mathbb{Z}$ is not torsion, the result follows from Theorem 4.1.12 above.

□

We note also that $F_1(G)$ for G an infinite torsion group must always contain non-regular languages (for example, the word problem of G). It follows that conditions (i) and (iv) in Proposition 4.1.11 are mutually exclusive.

The theorem is of particular interest because torsion groups which are not locally finite are rather rare and difficult to construct. Any locally finite group is certainly

torsion, however the converse is not true. The *Burnside problem* is one of the oldest and most famous questions in group theory, and remained unsolved for many decades. The Burnside problem in its original form is the following.

> If G is a torsion group (that is, all elements have finite order), and G is finitely generated, then is G necessarily a finite group?

The answer to this question was shown to be negative in 1964 by Golod and Shafarevich [25]. Variations on this original problem are still not entirely settled however.

It would be interesting to study the language classes $F_1(G)$ corresponding to particular known examples of infinite torsion groups [24, 27, 44].

4.2 Rational monoid automata

In Section 4.1, we saw that the extent to which traditional monoid automata can utilise the differences in structure between groups and monoids was limited. In this section, we consider a generalisation which allows us to make use of the full structure of arbitrary monoids. By removing the reliance on an identity element we are also able to consider more general semigroups.

Let S be a semigroup and Σ a finite alphabet. We define a *rational S-automaton* over Σ to be a finite automaton over the direct product $S \times \Sigma^*$ with a distinguished initial state, a set of distinguished terminal states, and two rational subsets $X_0, X_1 \subseteq S$ called the *initial set* and *terminal set* respectively. The automaton accepts a word $w \in \Sigma^*$ if there exists $x_0 \in X_0$ and $x \in S$ such that $x_0 x \in X_1$, and (x, w) labels a path from the initial state to a terminal state in the automaton. For S a semigroup, we let $F_{Rat}(S)$ denote the set of languages accepted by rational S-automata. We shall first deal with the case where S is a monoid.

The following proposition says that, for M a monoid, the initial set may be taken to be $\{1\}$ without loss of generality.

Proposition 4.2.1. *Let M be a monoid with identity 1, and $L \subseteq \Sigma^*$ a language. If $L \in F_{Rat}(M)$ is accepted by a rational M-automaton with initial set $X_0 \subseteq M$ and*

terminal set $X_1 \subseteq M$ then L is accepted by a rational M-automaton with initial set $\{1\}$ and terminal set X_1.

Proof. Let B be a rational monoid automaton with initial set $X_0 \subseteq M$ and terminal set $X_1 \subseteq M$ which accepts the language L. Since X_0 is a rational subset of M there exists some finite automaton over M which accepts X_0. Applying the map $x \mapsto (x, \epsilon)$ to the edge labels of that produces an automaton over $M \times \Sigma^*$ which we call A.

Now we construct a new M-automaton C with

- state set the disjoint union $Q = Q_A \cup Q_B$ where Q_A is the state set of A and Q_B is the state set of B;
- finite alphabet Σ;
- all of the edges of A and B;
- edges labelled $(1, \epsilon)$ connecting the terminal states of A to the initial state of B;
- initial state the initial state of A;
- terminal states the terminal states of B.

Then an accepting path through C consists of a path (x_0, ϵ) (with $x_0 \in X_0$) connecting the initial state to some terminal state of A, followed by an ϵ-transition, followed by a path connecting the initial state of B to a terminal state of B labelled by (x, w) with $x \in M$. Now a word $w \in \Sigma^*$ is accepted by the original automaton B precisely if there exists $x_0 \in X_0$ and an accepting path through B labelled by (x, w) such that $x_0 x \in X_1$. Hence $w \in L$ implies that w is accepted by C.

For the other implication, let w be accepted by C. Then we may break down the accepting path labelled by w in C as $(x_0, \epsilon)(x, w)$ for some $x_0, x \in M$ with $x_0 x \in X_1$ and (x, w) accepted by B. Moreover, by construction, $x_0 \in X_0$, and hence $w \in L$ as required.

□

Proposition 4.2.2. *Let M be a monoid. Then $F_1(M) \subseteq F_{Rat}(M)$.*

Proof. Since the set $\{1\}$ is a rational subset of M (accepted by the automaton with a single state and only one looped edge labelled by the identity element 1), an M-automaton with initial set $\{1\}$ and terminal set $\{1\}$ is a rational M-automaton. But this is precisely the usual definition of an M-automaton, and so the result follows. □

In the case that the register monoid is a group G, it transpires that rational G-automata are no more powerful than standard G-automata. This result is an obvious consequence of [19, Theorem 2.5] and Proposition 4.2.1 above, but for completeness and accessibility, we provide here a direct proof in the language of monoid automata.

Proposition 4.2.3. *Let G be a group. Then*

$$F_{Rat}(G) = F_1(G).$$

Proof. By Proposition 4.2.2, $F_1(G) \subseteq F_{Rat}(G)$. Conversely, suppose $L \in F_{Rat}(G)$. By Proposition 4.2.1 there is a rational G-automaton accepting L with initial set $\{1\}$.

Let A be some G-automaton with terminal rational subset X accepting the language L. Since X is a rational subset of G, by Proposition 2.2.6 the set $X^{-1} = \{x^{-1} \mid x \in X\}$ is also a rational subset of G. Consider a finite automaton accepting the set X^{-1}. The automaton has edges labelled by elements of G. By applying the map

$$\varphi : G \to G \times \Sigma^*, \quad g \mapsto (g, \epsilon)$$

to the edge labels we obtain a new automaton (over $G \times \Sigma^*$) such that every path from the initial state to some terminal state has label (x^{-1}, ϵ) for $x \in X$. We call this automaton B.

We construct a new G-automaton C with

- state set $Q = Q_A \cup Q_B$, the disjoint union of the state sets of A and B;
- finite alphabet Σ;
- all of the edges of A and B;

- edges labelled $(1, \epsilon)$ connecting the terminal states of A to the initial state of B;
- initial state the initial state of A;
- terminal states the terminal states of the automaton B.

Let w be accepted by C. Then an accepting path labelled by w may be factorised $(x_1, w_1)(1, \epsilon)(x_2, w_2)$ with $x_1x_2 = 1 \in G$ and $w = w_1w_2$. Then $x_2 = x_1^{-1}$ and $x_1 \in X$ and $x_2 = x_1^{-1} \in X^{-1}$. Since all edges from B have righthand label ϵ, $w = w_1$ and hence $w \in L$.

Conversely, let $w \in L$. Then there exists an accepting path in A with label (x, w) with $x \in X$. Thene $x^{-1} \in X^{-1}$ is such that $xx^{-1} = 1$ and hence there exists an accepting path through B labelled by x^{-1}. By construction there exists an accepting path through C with label $(x1x^{-1}, w)$ and hence w is accepted by C.

So the automaton C constructed in this way is in fact a G-automaton accepting the language L, which completes the proof. □

A number of more general results about semigroups will prove useful later. Though a rational subset K of a semigroup S is certainly the homomorphic image of a regular language, the full pre-image of K in the free monoid need not be regular. It is this observation which informs the following result.

Proposition 4.2.4. *Let $\sigma : X^+ \to S$ be a finite choice of generators for a semigroup S. If $K \subseteq S$ is a subset of S such that $K' = \{w \in X^+ \mid w\sigma \in K\}$ is regular and $R \subseteq S$ is a rational subset then $R \setminus K$ is also a rational subset of S.*

Proof. Since $R \subseteq S$ is a rational subset, there exists some regular language $L \subseteq X^+$ such that $L\sigma = R$. Clearly $L \setminus K'$ is regular, and since $(L \setminus K')\sigma = R \setminus K$ the result follows. □

In the special case of a semigroup with a zero element, we have the following obvious consequence.

Corollary 4.2.5. *Let $\sigma : X^+ \to S$ be a finite choice of generators for a semigroup S with zero. If $\{w \in X^+ \mid w\sigma = 0\}$ is regular and $R \subseteq S$ is a rational subset then $R \setminus \{0\}$ is also a rational subset of S.*

Proposition 4.2.6. *Let S be a semigroup without zero and $\sigma : X^+ \to S^0$ be a finite choice of generators for S^0. Then the set $\{z \in X^+ \mid z\sigma = 0\}$ is regular.*

Proof. Let $Z = \{x \in X \mid x\sigma = 0\}$. Since S is a subsemigroup of S^0, a word $w \in X^+$ such that $w\sigma = 0$ must contain some $z \in Z$. Then the set

$$\{z \in X^+ \mid z\sigma = 0\} = \bigcup_{z \in Z} X^* z X^*$$

is regular. □

4.3 Transductions and closure properties

In this section we study the relationship between rational transductions and rational monoid and semigroup automata. We have already considered rational transductions for regular languages (Section 2.2), and noted the generalisation of these results to traditional monoid automata (Proposition 3.0.2). We now give an analogous result for rational monoid automata.

Proposition 4.3.1. *Let M be a monoid, $L \subset \Sigma^*$ a language and $X \subseteq M$ a subset. Then the following are equivalent.*

(i) L is accepted by an M-automaton with initial set $\{1\}$ and target set X;

(ii) there exists an alphabet Ω and a morphism $\omega : \Omega^* \to M$ such that L is a rational transduction of $X\omega^{-1}$.

If M is finitely generated then the following condition is also equivalent to those above.

(iii) For every choice of generators $\omega : \Omega^* \to M$ for M, L is a rational transduction of $X\omega^{-1}$.

Proof. The proof follows the same pattern as Proposition 3.0.2.

To show that (i) implies (ii), suppose L is accepted by an M-automaton with initial set $\{1\}$ and target set X. Choose a finite alphabet Ω and a map $\omega : \Omega^* \to M$ such that the image $\Omega^*\omega$ contains every element of M which forms the first component of an edge label in the automaton. We now obtain from the automaton a transducer over $\Omega^* \times \Sigma^*$ by replacing each edge label (m, x) with (w, x) where $w \in \Omega^*$ is some word such that $w\omega = m$.

Now $w \in L$ if and only if there is a path connecting an initial state in the M-automaton to a terminal state labelled by (m, w) for some $m \in X$. But this holds if and only if there exists a path through the transducer defined above labelled by (x, w) for some $x \in X\omega^{-1}$. Hence L is the image under a rational transduction of the set $X\omega^{-1}$ as required.

To show that (ii) implies (i), suppose we are given a map $\omega : \Omega^* \to M$, and a rational transducer such that L is the image of $X\omega^{-1}$ under the transduction. We construct an M-automaton from the transducer by replacing edge labels of the form $(x\omega^{-1}, w)$ with (x, w), taking initial set $\{1\}$ and terminal set X. We show that the automaton constructed in this way accepts precisely the language L.

Let $w \in L$. Then there exists a path through the transducer labelled by $(x\omega^{-1}, w)$ where $x \in X$. So there exists a path through the automaton labelled by (x, w), and hence w is accepted by the automaton. Conversely, suppose $w \in \Sigma^*$ is accepted by the M-automaton constructed above. Then there exists a path through the automaton labelled (x, w) for some $x \in X$. Hence there exists a path through the transducer labelled by $(x\omega^{-1}, w)$ and hence $w \in L$ as required.

Suppose now that M is finitely generated. Clearly, (iii) implies (ii). Finally, if (ii) holds then we can extend ω arbitrarily to a finite choice of generators $\omega' : (\Omega')^* \to M$. Since under this choice of generators, the labelling of the rational transducer will not change, the desired property follows as above, so that (iii) holds. □

Proposition 4.3.2 below gives a characterisation of classes of languages accepted by M-automata with rational target sets in terms of rational subsets and transductions.

Proposition 4.3.2. *Let M be a monoid and $L \subseteq \Sigma^*$ a language. Then the following are equivalent.*

(i) $L \in F_{Rat}(M)$*;*

(ii) there exists a finite alphabet Ω, a morphism $\omega : \Omega^* \to M$ and a rational subset $X \subseteq M$ such that L is a rational transduction of $X\omega^{-1}$.

If M is finitely generated then the following condition is also equivalent to those above.

(iii) There exists a rational subset $X \subseteq M$ such that for every finite choice of generators $\omega : \Omega^* \to M$ for M, L is a rational transduction of $X\omega^{-1}$.

Proof. Let $L \subseteq \Sigma^*$ be a language satisfying (i). Then by Proposition 4.2.1 there exists a rational M-automaton with initial set $\{1\}$ and rational target set $X \subseteq M$ accepting L. Then L satisfies property (i) of Proposition 4.3.1 from which it follows that (ii) holds. The other implications are proven similarly. □

Proposition 4.3.3. *$F_{Rat}(M)$ is a rational cone. In particular, it is closed under morphism, inverse morphism, intersection with regular languages, and (since it contains a non-empty language) union with regular languages.*

Proof. Since rational transductions are closed under composition (Theorem 2.2.10) we conclude that $F_{Rat}(M)$ consists precisely of languages $L \subseteq \Sigma^*$ which are rational transductions of rational subsets of M by Proposition 4.3.2 above. Hence $F_{Rat}(M)$ is closed under rational transductions and the result follows immediately from Theorem 2.2.8. □

Next we wish to widen the scope of our study to semigroups. We first need a more general definition.

Let $X_0, X_1 \subseteq S$ be subsets of a semigroup S. Then their *set difference* is the set

$$X_0^{-1}X_1 \;=\; \{x \in S \mid x_0 x = x_1 \text{ for some } x_0 \in X_0, x_1 \in X_1\}.$$

We say that a subset $X \subseteq S$ is a *rational set difference* if there exist rational subsets $X_0, X_1 \subseteq S$ such that $X = X_0^{-1}X_1$. Note that in a group, the rational set differences are exactly the rational subsets, but in a general semigroup this does not hold.

The following statement is a semigroup analogue of Proposition 4.3.1. Since we are working with semigroups which do not necessarily have identity elements, relations in the Proposition below take the form $\rho \subseteq \Omega^+ \times \Sigma^*$.

Proposition 4.3.4. *Let X_0 and X_1 be subsets of a semigroup S, and let $L \subseteq \Sigma^*$ be a language. Then the following are equivalent:*

(i) L is accepted by an S-automaton with initial set X_0 and terminal set X_1;

(ii) there exists a finite alphabet Ω, a morphism $\omega : \Omega^+ \to S$ and a rational relation $\rho \subseteq \Omega^+ \times \Sigma^*$ such that

$$L = (X_0^{-1}X_1)\omega^{-1}\rho.$$

If S is finitely generated then the following condition is also equivalent to those above.

(iii) For every finite choice of generators $\omega : \Omega^+ \to S$ for S, there exists a rational relation $\rho \subseteq \Omega^+ \times \Sigma^*$ such that

$$L = (X_0^{-1}X_1)\omega^{-1}\rho.$$

Proof. The proof is similar to the proof of Proposition 4.3.2. To show that (i) implies (ii), suppose that L is accepted by an S-automaton A with initial set X_0 and terminal set X_1. Choose a finite alphabet Ω and a map $\omega : \Omega^+ \to S$ such that the image $\Omega^+\omega$ contains every element of S which forms the first component of an edge label in the automaton. We now obtain from A a finite automaton B over $\Omega^+ \times \Sigma^*$ by replacing each edge label (s, x) with (w, x) for some $w \in \Omega^+$ such that $w\omega = s$. The automaton resulting from this change of labelling defines a rational relation $\rho \subseteq \Omega^+ \times \Sigma^*$.

Let $X \subseteq S$ denote the set of elements of S labelling paths through A connecting the initial state to a terminal state. Then a word $w \in \Sigma^*$ is accepted by A if and only if there exists a path from the initial state to a terminal state labelled by (x, w) for some $x \in X \cap X_0^{-1}X_1$. So let $w \in L$ with accepting path (x, w) for some $x \in S$. Then $x \in X_0^{-1}X_1$, and there exists a path through B labelled by $(y, w) \in \Omega^+ \times \Sigma^*$ such that $y\omega = x$. So $(y, w) \in \rho$ and $w = (x\omega^{-1})\rho$ as required. The converse is similar.

For (ii) implies (i), suppose we are given a map $\omega : \Omega^+ \to S$ and an automaton B over $\Omega^+ \times \Sigma^*$ such that L is the image under the relation accepted by B (that is, ρ) of the language $(X_0^{-1}X_1)\omega^{-1}$. We construct from B a new automaton A over $S \times \Sigma^*$ by applying the map ω to the first component of each edge label. Considering A as an S-automaton with initial set X_0 and terminal set X_1, we let $(x, w) \in \Omega^+ \times \Sigma^*$ label an accepting path through the automaton B. On applying ω we obtain an accepting path through the automaton A of the form $(s, w) \in S \times \Sigma^*$. Now since $s \in X_0^{-1}X_1$, there exists some $x_0 \in X_0$ such that $x_0 s \in X_1$ and so $w \in L$, hence A accepts the language L. Again the converse is similar.

Suppose now that S is finitely generated. Clearly (iii) implies (ii). Conversely, if (ii) holds then we may extend ω arbitrarily to a finite choice of generators $\omega' : (\Omega')^+ \to S$. Since $\Omega \subseteq \Omega'$, we may consider the rational relation ρ as a rational relation over $(\Omega')^+ \times \Sigma^*$ and so $L = (X_0^{-1}X_1)\omega^{-1}\rho$ and (iii) holds. □

As a corollary, we immediately obtain the following characterisation for language classes of the form $F_{Rat}(S)$.

Corollary 4.3.5. *Let S be a semigroup and $L \subset \Sigma^*$ a language. Then the following are equivalent.*

(i) $L \in F_{Rat}(S)$;

(ii) there exists an alphabet Ω, a morphism $\omega : \Omega^+ \to S$, a rational set difference $X \subseteq S$ and a rational relation $\rho \subseteq \Omega^+ \times \Sigma^*$ such that $L = X\omega^{-1}\rho$.

If S is finitely generated then the following condition is also equivalent to those above.

(iii) There exists a rational set difference $X \subseteq S$ such that for every finite choice of generators $\omega : \Omega^+ \to S$ for S, there exists a rational relation $\rho \subseteq \Omega^+ \times \Sigma^*$ such that $L = X\omega^{-1}\rho$.

Proof. Let $L \subseteq \Sigma^*$ be a language satisfying (i). Then there exists a rational S-automaton A with rational initial and terminal sets X_0 and X_1 respectively accepting L. Let $X = X_0^{-1}X_1$. Then L satisfies (i) in Proposition 4.3.4 from which it follows that (ii) holds. The other implications are proved similarly. □

Note that, unlike in the monoid case, we cannot conclude that $F_{Rat}(S)$ is a rational cone. This is because the composition of a rational relation in $\Omega^+ \times \Sigma^*$ with a rational transduction from Σ^* to another free monoid Γ^* need not be a rational relation in $\Omega^+ \times \Gamma^*$ (although it will be rational in $\Omega^* \times \Gamma^*$).

4.4 Adjoining a zero

In this section we consider the operation of adjoining a zero to a given monoid, and how this may affect the language classes corresponding to the resulting monoids.

We shall need the following simple result.

Theorem 4.4.1. *Let M be a monoid. Then $F_{Rat}(M^0) = F_{Rat}(M)$.*

Proof. Let $L \subseteq \Sigma^*$ be accepted by a rational M-automaton with initial set X_0 and terminal set X_1. Since $M \subset M^0$, it is clear that the rational sets X_0 and X_1 are rational subsets of M^0. Similarly, for every $m \in M$ labelling an edge in the automaton it is clear that $m \in M^0$. Thus the automaton is also a rational M^0-automaton and accepts precisely the language $L \subseteq \Sigma^*$. Hence we conclude that $F_{Rat}(M) \subset F_{Rat}(M^0)$.

Conversely, suppose $L \in F_{Rat}(M^0)$. Then by Proposition 4.2.1 we may choose a rational M^0-automaton A accepting L with initial set $\{1\}$. Let $X_1 \subseteq M$ be the terminal set of the automaton A.

Let L_0 be the language of words $w \subseteq \Sigma^*$ such that $(0, w)$ labels a path from the initial state to a terminal state. Let L_1 be the set of words w such that (m, w) labels a path from the initial state to a terminal state for some $m \in X_1 \setminus \{0\}$. Clearly either $L = L_0 \cup L_1$ (in the case that $0 \in X_1$) or $L = L_1$ (if $0 \notin X_1$). We claim that L_0 is regular and that $L_1 \in F_{Rat}(M)$. By Proposition 4.3.3 this will suffice to complete the proof.

The argument to show that $L_1 \in F_{Rat}(M)$ is very similar to the proof of Proposition 4.1.3. We construct from the rational M^0-automaton A a new rational M-automaton B by simply removing each edge which has a label of the form $(0, m)$. The new automaton B has initial set $\{1\}$ and terminal set $X_1 \setminus \{0\}$. It is straightforward to show, using exactly the same techniques as in Proposition 4.1.3, that B

accepts exactly the language L_1.

It remains to prove that L_0 is regular. Let Q be the vertex set of the automaton A, and let $Q_0 = \{q_0 \mid q \in Q\}$ and $Q_1 = \{q_1 \mid q \in Q\}$ be disjoint copies of Q. We define from A a finite automaton C with

- state set $Q_0 \cup Q_1$;
- for each edge in A from p to q with label of the form (m, x) $(m \neq 0)$
  - an edge from p_0 to q_0 labelled x and
  - an edge from p_1 to q_1 labelled x;
- for each edge in A from p to q with label of the form $(0, x)$
  - an edge from p_0 to q_1 labelled x and
  - an edge from p_1 to q_1 labelled x;
- initial state q_0 where q is the initial state of A; and
- terminal states q_1 whenever q is a terminal state of A.

We claim that C accepts exactly the set L_0. Let $w \in L_0$. Then there exists an accepting path π through A labelled $(0, w)$. It follows from the definition of M^0 that no product of non-zero elements can equal 0; hence, this path must traverse at least one edge labelled $(0, x)$ for some $x \in \Sigma^*$. Suppose then that $\pi = \pi_1\pi_2\pi_3$ where π_1 is a path from the initial vertex to a vertex p with label (m_1, w_1), π_2 is the first edge in the path encountered with label 0, an edge from p to a vertex q with label $(0, x)$ say, and π_3 is a path from q to a terminal vertex with label (m_3, w_3). Then there exists a path in C from the initial vertex to p_0 labelled w_1, an edge from p_0 to q_1 with label x, and an edge from q_1 to a terminal vertex with label w_3. Hence, $w = w_1xw_3$ is accepted by C, as required.

Conversely suppose $w \in L(C)$, and let π be an accepting path for w. Notice that the initial vertex of C lies in Q_0 while all the terminal vertices lie in Q_1. Then $\pi = \pi_1\pi_2\pi_3$ where π_1 is a path from the initial vertex to some p_0 with label w_1, π_2

is an edge from p_0 to some q_1 with label x, π_3 is a path from q_1 to a terminal vertex with label w_3 where $w = w_1 x w_3$. It follows easily from the definition of C that there exists a path in A from the initial vertex to p with label of the form (m_1, w_1); an edge from p to q with label $(0, x)$ and a path from q to a terminal vertex with label of the form (m_3, w_3). Thus, A accepts $(m_1 0 m_3, w_1 x w_3) = (0, w)$ so that $w \in L_0$ as required. □

To prove a similar result in the general case of semigroups, we require a few more results.

Lemma 4.4.2. *Let $\sigma : X^+ \to S$ be a finite choice of generators for a semigroup S and let $X_0, X_1 \subseteq S$ be such that there exist words $w \in X^+$ of arbitrarily high length with $w\sigma \in X_0^{-1} X_1$. Then the set of languages accepted by S-automata with initial set X_0 and terminal set X_1 contains the regular languages.*

Proof. Let A be a finite automaton accepting a regular language $L \subseteq \Sigma^*$. We construct a new S-automaton B from A with initial set X_0 and terminal set X_1 with

- state set the state set of A plus a new state q_t which will be the unique terminal state;
- for each edge connecting a state p to a state q in A labeled by $w \in \Sigma^*$ and for each $x \in X$ an edge from the state p to the state q in B labeled by $(x\sigma, w)$;
- for each terminal state q in A and each $x \in X$ an edge from the state q to the state q_t in B labeled by $(x\sigma, \epsilon)$;
- the initial states of the S-automaton B will be the same as the initial states of A and
- for each $x \in X$ we also add a loop at the state q_t labeled by $(x\sigma, \epsilon)$.

Since the automaton A and the set X are finite the edge set of B will be finite; we claim the S-automaton constructed in this way accepts precisely the language L.

Let $w \in \Sigma^*$ be accepted by B. Then since every edge in B with righthand edge label not equal to ϵ has an equivalent edge in A with the same label from Σ^* we may easily conclude that $w \in L$.

Conversely let $w \in L$ with $|w| = n$. Let $x \in X^+$ be such that $x = x_1 \dots x_k$ with $k > n$ and $x\sigma \in X_0^{-1}X_1$. Then there exists a path in B connecting the initial state to some state q such that q was a terminal state in A labeled by $((x_1 \dots x_n)\sigma, w)$ followed by an edge from q to the terminal state q_t labeled by $((x_{n+1})\sigma, \epsilon)$. There also exists a closed path at q_t labeled by $((x_{n+2} \dots x_k)\sigma, \epsilon)$ and so since $x \in X_0^{-1}X_1$ we may conclude that w is accepted by the S-automaton B with initial set X_0 and terminal set X_1 as required. □

Lemma 4.4.3. *Let $\sigma : X^+ \to S$ be a finite choice of generators for a semigroup S. Let $X_0, X_1 \subseteq S$ be such that there is an upper bound on the length of words $w \in X^+$ such that $w\sigma \in X_0^{-1}X_1$. Then every language $L \subseteq \Sigma^*$ accepted by an S-automaton with initial set X_0 and terminal set X_1 must be finite.*

Proof. Suppose for a contradiction that A is an S-automaton with initial set X_0 and terminal set X_1 such that X_0 and X_1 satisfy the condition above, and the language $L \subseteq \Sigma^*$ accepted by A is infinite. Since L is infinite A must contain paths of arbitrary length connecting the initial state to some terminal state. For each $s \in S$ labelling an edge in A let $w_s \in X^+$ be a word such that $w_s\sigma = s$. For each path π in A let $w_\pi = w_{s_1}w_{s_2} \dots w_{s_n}$ where each s_i denotes a successive edge label from S on the path. If a path π is accepting then $w_\pi\sigma \in X_0^{-1}X_1$ and clearly $|w_\pi|$ is greater than or equal to the number of edges in the path. So the set of all words w_π such that π is an accepting path contains words of arbitrary length such that $w_\pi \in X_0^{-1}X_1$ giving the required contradiction. □

Proposition 4.4.4. *For any semigroup S the family of languages $F_{Rat}(S)$ contains the regular languages.*

Proof. Note first that S may not be finitely generated. Choose any finitely generated subsemigroup S' of S and let $X_0 = X_1 = S'$. By Lemma 4.4.2 for any regular language

$L \subseteq \Sigma^*$ we may construct an S-automaton with initial and terminal set S accepting L and hence the regular languages are contained in $F_{Rat}(S)$ as required. □

Theorem 4.4.5. *For S a finitely generated semigroup the language family $F_{Rat}(S)$ is closed under union with regular languages.*

Proof. Let $\sigma : X^+ \to S$ be a finite choice of generators for S and let $L \subseteq \Sigma^*$ be a language in $F_{Rat}(S)$ accepted by a rational S-automaton with initial set $X_0 \subseteq S$ and terminal set $X_1 \subseteq S$. We let $K \subseteq \Sigma^*$ be a regular language. Now if the set $X_0^{-1}X_1$ contains the image under σ of words of arbitrarily long length over the generating set X, by Lemma 4.4.2 we may construct a rational S-automaton B with initial set X_0 and terminal set X_1 accepting precisely the language K. It remains to show that we may 'merge' the S-automata A and B to form a rational S-automaton C with the same initial and terminal set as A and B accepting precisely the union $L \cup K$. The format of the proof is similar in spirit to the proof of Proposition 2.2.4.

We construct the S-automaton C from A and B. Let Q_A and Q_B denote the state sets of A and B respectively, and let q_0 and q_0' denote the initial states of A and B. C has:

- state set the disjoint union $Q_A \cup Q_B$;
- a single initial state q having all of the outgoing edges of both q_0 and q_0';
- all other edges of A and B;
- terminal state set $F_A \cup F_B$ where F_A is the set of terminal states of A and F_B is the set of terminal states in B.

A straightforward argument allows us to conclude that the S-automaton constructed in this way accepts precisely the union $L \cup K$ as required.

It remains to deal with the case that the set $X_0^{-1}X_1$ contains only words $w\sigma$ where w is of length smaller than some upper bound. In this case by Lemma 4.4.3 the language L must be finite and hence regular. So the union $L \cup K$ is again a regular language and we may apply Proposition 4.4.4 to conclude that $L \cup K$ is contained in $F_{Rat}(S)$ as required. This completes the proof. □

We are finally in a position to prove a semigroup analogue of Theorem 4.4.1.

Theorem 4.4.6. *For S a finitely generated semigroup, $F_{Rat}(S) = F_{Rat}(S^0)$.*

Proof. Let A be a rational S-automaton with initial set X_0 and terminal set X_1 accepting the language $L \subseteq \Sigma^*$. Since $S \subset S^0$ it is clear that X_0 and X_1 are contained in S^0 and that each $s \in S$ labelling an edge in A is also contained in S^0. Hence we may conclude that $F_{Rat}(S) \subseteq F_{Rat}(S^0)$ and so we need only prove the converse.

Let A be a rational S^0-automaton with initial set X_0 and terminal set X_1 accepting the language $L \subseteq \Sigma^*$. If 0 is not contained in either X_0 or X_1 then a path labeled by zero can never be accepting and we may consider A as a rational S-automaton and clearly $L \subseteq F_{Rat}(S)$.

If $0 \in X_0$ but $0 \notin X_1$ then $\{0\}^{-1}X_1 \subset X_0^{-1}X_1$ is empty and it is easily seen that by taking the initial set to be $X_0 \setminus \{0\}$ we may consider A as a rational S-automaton and $L \subseteq F_{Rat}(S)$ as required. Note that since the zero is adjoined to S, Proposition 4.2.6 says that the set of words over the generators of S which are mapped to zero under σ is regular, and hence by Corollary 4.2.5 if the set X_0 is rational then so is $X_0 \setminus \{0\}$.

If $0 \in X_0$ and $0 \in X_1$ then L consists of all words $w \in \Sigma^*$ such that (x, w) labels a path connecting the initial state of A to some terminal state of A for some $x \in S$. Hence L is regular and by Proposition 4.4.4 is contained in $F_{Rat}(S)$ as required.

The final case to consider occurs when $0 \notin X_0$ and $0 \in X_1$. Clearly we may write $L = L_0 \cup L_1$ where L_1 is accepted by an S-automaton with 0 not in the initial or terminal sets and L_0 is accepted by an S^0-automaton with initial and terminal set $\{0\}$. Applying the same methods as in the proof of Theorem 4.4.1 we conclude that L_0 is regular and $L_1 \in F_{Rat}(S)$, and by Theorem 4.4.5 this suffices to complete the proof.

□

Finally, we turn our attention to the case of groups. Combining Proposition 4.2.3 and Theorem 4.4.1 gives us the following immediate corollary.

Corollary 4.4.7. *Let G be a group. Then $F_{Rat}(G^0) = F_1(G)$.*

Proof. Let L be a language accepted by the rational G^0-automaton A with initial set X_0 and terminal set X_1. Then by Theorem 4.4.1 there exists a rational G-automaton B accepting the same language L. Now Proposition 4.2.3 says that there exists a G-automaton with initial and terminal sets equal to the identity element accepting L. Hence $L \in F_1(G)$.

The converse is clear.

□

Chapter 5

Polycyclic monoids

In this chapter we turn our attention to the classes $F_{Rat}(P(X))$ of languages accepted by polycyclic monoid automata with rational target sets. Recall that the polycyclic monoids form the natural algebraic model of pushdown stores. Some of the material in this chapter has been published in [49, 50].

For $|X| \geq 2$, it transpires that every language accepted by a $P(X)$-automaton with rational target set is accepted by a $P(X)$-automaton, and hence that $F_{Rat}(P(X))$ is the class of context-free languages. In order to prove this, we will need some results about rational subsets of polycyclic monoids, which we establish using techniques from string rewriting theory. These results may be of independent interest.

5.1 The structure of rational subsets

In this section we consider a normal form for elements of polycyclic monoids, and how this form affects the structure of the rational subsets. We begin with a definition.

A *monadic string rewriting system* Λ over an alphabet Σ is a subset of $\Sigma^* \times \{\Sigma \cup \{\epsilon\}\}$. We normally write an element $(w, x) \in \Lambda$ as $w \to x$. Then we write $u \Rightarrow v$ if $u = rws \in \Sigma^*$ and $v = rxs \in \Sigma^*$ with $w \to x$. Denote by $\Rightarrow^*$ the transitive, reflexive closure of the relation $\Rightarrow$. If $u \Rightarrow^* v$ we say that u is an *ancestor* of v under Λ and v is a *descendant* of u under Λ; we write $L\Lambda$ for the set of all descendants of words in L. The set of words which cannot be reduced any further under the rewriting system

Λ are called the Λ-*irreducible words*.

We note that the image of any regular set under a finite monadic string rewriting system will again be a regular set [8], a useful property which we shall use in the sequel. For more information on such systems see [7, 8].

Theorem 5.1.1. *Let X be a finite alphabet and R a rational subset of $P^0(X)$, and let $\sigma : \Sigma_X^* \to P^0(X)$ be a finite choice of generators. Then there exists a regular language*

$$L \subseteq Q_X^* P_X^* \cup \{z\}$$

such that $L\sigma = R$. Moreover, there is an algorithm which, given an automaton recognizing a regular language $G \subseteq \Sigma_X^*$, constructs an automaton recognising a language $L \subseteq Q_X^* P_X^* \cup \{z\}$ with $L\sigma = G\sigma$.

Proof. Since R is rational, there exists a regular language $K \subseteq \Sigma_X^*$ such that $K\sigma = R$. We define a monadic rewriting system Λ on Σ_X^* with the following rules:

$$p_x q_x \to \epsilon, \qquad p_x q_y \to z, \qquad z q_x \to z,$$
$$p_x z \to z, \qquad z p_x \to z, \qquad q_x z \to z,$$
$$zz \to z$$

for all $x, y \in X$ with $x \neq y$.

Note that the only combination of two letters which is not featured in the rewriting rules above is of the form $q_x p_y$ for $x, y \in X$. We may conclude then that the language of Λ-irreducible words is exactly $Q_X^* P_X^* \cup \{z\}$, since the rewriting rules reduce all other letter combinations to z or the empty word. With this in mind, we define

$$L \;=\; K\Lambda \cap (Q_X^* P_X^* \cup \{z\})$$

Certainly L is regular, and moreover an automaton for L can be effectively computed from an automaton for K. Thus, it will suffice to show that $L\sigma = R$.

By definition $L\sigma \subseteq (K\Lambda)\sigma$, and since the rewriting rules are all relations satisfied in $P^0(X)$,

$$(K\Lambda)\sigma \;\subseteq\; K\sigma \;=\; R.$$

Conversely, if $s \in R$ then $s = w\sigma$ for some $w \in K$. Since the rules of Λ are all length-reducing w must have an irreducible descendant, say w'. But now $w' \in L$ and $w'\sigma = w\sigma = s$ so that $s \in L\sigma$. Thus, $L\sigma = R$ as required.

□

As a corollary we obtain a corresponding result for bicyclic monoids.

Corollary 5.1.2. *Let R be a rational subset of a bicyclic monoid B, and $\sigma : \{p, q\}^* \to B$ the natural surjective morphism. Then there exists a regular language $L \subseteq q^*p^*$ such that $L\sigma = R$. Moreover, there is an algorithm which, given an automaton recognizing a regular language $G \subseteq \{p, q\}^*$, constructs an automaton recognising a language $L \subseteq q^*p^*$ with $L\sigma = G\sigma$.*

Proof. Let $R \subseteq B$ be a rational subset of the bicyclic monoid. Since B is a submonoid of B^0, R is a rational subset of B^0. We extend σ to $\sigma' : \{p, q, z\}^* \to B^0$ by setting $z\sigma' = 0$. By Theorem 5.1.1, there exists a regular language of the form $L \subseteq q^*p^* \cup \{z\}$ such that $L\sigma = R$. But $z \notin L$ since $z\sigma' = 0 \notin R$ and so $L \subseteq q^*p^*$ and $L\sigma = R$ as required. Moreover, the automaton for L can be effectively computed. □

Before proceeding to apply the theorem to polycyclic monoid automata with rational target sets, we note some general consequences of Theorem 5.1.1 for rational subsets of polycyclic monoids. A collection of subsets of a given *base set* is called a *boolean algebra* if it is closed under union, intersection and complement within the base set.

Corollary 5.1.3. *The rational subsets of any finitely generated polycyclic monoid form a boolean algebra. Moreover, the operations of union, intersection and complement are effectively computable.*

Proof. Let L and K be rational subsets of a finitely generated polycyclic monoid. Then there exist finite automata A and B over the monoid accepting L and K respectively. By Proposition 2.2.5 we may conclude that the set of rational subsets of a finitely generated polycyclic monoid is effectively closed under union.

Let $\overline{L}$ denote the complement of a set L. Then we may describe intersection in terms of union and complement as follows:

$$L \cap K = \overline{\overline{L} \cup \overline{K}}$$

where L and K are sets. Hence it suffices to show that the rational subsets of polycyclic monoids are closed (effectively) under complement. To this end, suppose first that R is a rational subset of a finitely generated polycyclic monoid $P(X)$ with $|X| \geq 2$ so that $P(X) = P^0(X)$. Then by Theorem 5.1.1, there is a regular language $L \subseteq (Q_X^* P_X^* \cup \{z\})$ such that $L\sigma = R$. Let $K = (Q_X^* P_X^* \cup \{z\}) \setminus L$. Then K is regular and, since $Q_X^* P_X^* \cup \{z\}$ contains a unique representative for every element of $P(X)$, it is readily verified that $K\sigma = P(X) \setminus (L\sigma)$. Thus, $P(X) \setminus (L\sigma)$ is a rational subset of $P(X)$, as required.

For effective computation of complements, observe that given an automaton recognizing a language $R \subseteq \Sigma_X^*$, we can by Theorem 5.1.1 construct an automaton recognizing a regular language $L \subseteq (Q_X^* P_X^* \cup \{z\})$ with $L\sigma = R\sigma$. Clearly we can then compute the complement $K = (Q_X^* P_X^* \cup \{z\}) \setminus L$ of L in $(Q_X^* P_X^* \cup \{z\})$, and since $K\sigma = P(X) \setminus (L\sigma)$, this suffices.

In the case that $|X| = 1$, the statement can be proved in a similar way but using Corollary 5.1.2 in place of Theorem 5.1.1. □

As another corollary, we obtain the decidability of the rational subset problem for finitely generated polycyclic monoids.

Corollary 5.1.4. *Finitely generated polycyclic monoids have decidable rational subset problem.*

Proof. Let $|X| \geq 2$ [respectively, $|X| = 1$]. Suppose we are given a rational subset R of $P(X)$ (specified as an automaton over Σ_X^* [respectively $\{p, q\}^*$]) and an element w (specified as a word in the appropriate alphabet). Clearly, we can compute $\{w\} \subset P(X)$ as a regular language. Indeed, let $|w| = n$. Then we construct an automaton with $n + 1$ states (labelled 1 to $n + 1$) and n edges. We let state 1 be the initial state, and state $n + 1$ be the single terminal state. Then state i is connected to state

$i+1$ via an edge labelled by the ith letter of w. By Corollary 5.1.3 we can compute a regular language $K \subseteq \Sigma_X^*$ [respectively, $\{p,q\}^*$] such that $K\sigma = R \cap \{w\}\sigma$. So $w\sigma \in R$ if and only if $R \cap \{w\}\sigma$ is non-empty, that is, if and only if K is non-empty. Since emptiness of regular languages is testable (Proposition 2.2.1), this completes the proof. □

Before returning to our main task of proving that $F_{Rat}(M) = F_1(M)$ for M a polycyclic monoid of rank 2 or more, that is, that polycyclic monoid automata with rational target sets accept only context-free languages, we need some more preliminary results.

Corollary 5.1.5. *Let R be a rational subset of $P^0(X)$ and suppose that $0 \notin R$. Then there exists an integer n and regular languages $Q_1, \ldots, Q_n \subseteq Q_X^*$ and $P_1, \ldots, P_n \subseteq P_X^*$ such that*

$$R = \bigcup_{i=1}^{n} (Q_i P_i)\,\sigma.$$

Proof. By Theorem 5.1.1, there is a regular language $L \subseteq Q_X^* P_X^*$ such that $L\sigma = R$. Let A be a finite automaton accepting L, with vertices numbered $1, \ldots, n$. Suppose without loss of generality that the edges in A are labelled by single letters from $Q_X \cup P_X$. For each i let Q_i be the set of all words in Q_X^* which label paths from the initial vertex to vertex i. Similarly, let P_i be the set of all words in P_X^* which label words from vertex i to a terminal vertex. It is easily seen that Q_i and P_i are regular.

Now if $w \in Q_i P_i$ then $w = uv$ where $u \in Q_X^*$ labels a path from the initial vertex to vertex i, and $v \in P_X^*$ labels a path from vertex i to a terminal vertex. Hence $uv = w$ labels a path from the initial vertex to a terminal vertex, and so $w \in L$. Conversely, if $w \in L \subseteq Q_X^* P_X^*$ then w admits a factorisation $w = uv$ where $u \in Q_X^*$ and $v \in P_X^*$. Since the edge labels in A are single letters, an accepting path for w must consist of a path from the initial vertex to some vertex i labelled u, followed by a path from i to a terminal vertex labelled v. It follows that $u \in Q_i$ and $v \in P_i$, so that $w \in Q_i P_i$. Thus we have

$$L = \bigcup_{i=1}^{n} Q_i P_i$$

and so

$$R \;=\; L\sigma \;=\; \left(\bigcup_{i=1}^{n} Q_i P_i\right)\sigma \;=\; \bigcup_{i=1}^{n} (Q_i P_i)\,\sigma$$

as required. □

For the next proposition, we will need some notation. For a word $q = q_{x_1} q_{x_2} \dots q_{x_n} \in Q_X^*$, we let $q' = p_{x_n} \dots p_{x_2} p_{x_1} \in P_X^*$. Similarly for a word $p = p_{x_1} p_{x_2} \dots p_{x_n} \in Q_X^*$, we let $p' = q_{x_n} \dots q_{x_2} q_{x_1} \in Q_X^*$. Note that $p'' = p$ and $q'' = q$. Note also that $p'\sigma$ is the unique *right inverse* of $p\sigma$, and $q'\sigma$ is the unique *left inverse* of $q\sigma$. Recall that a right [respectively, left] inverse of an element $a \in M$ is an element $b \in M$ such that $ab = 1$ [respectively, $ba = 1$].

Proposition 5.1.6. *Let $u \in \Sigma_X^*$, and let $q \in Q_X^*$ and $p \in P_X^*$. Then $u\sigma = (qp)\sigma$ if and only if there exists a factorisation $u = u_1 u_2$ such that $(q'u_1)\sigma = 1 = (u_2 p')\sigma$.*

Proof. Suppose first that $u\sigma = (qp)\sigma$. Let Λ be the monadic rewriting system defined in the proof of Theorem 5.1.1. Then u is reduced by Λ to qp. Notice that the only rules in Λ which can be applied to words not representing zero remove factors representing the identity; it follows easily that u admits a factorisation $u = u_1 u_2$ where $u_1\sigma = q\sigma$ and $u_2\sigma = p\sigma$. Now we have

$$(q'u_1)\sigma = (q'\sigma)(u_1\sigma) = (q'\sigma)(q\sigma) = 1$$

and symmetrically

$$(u_2 p')\sigma = (u_2\sigma)(p'\sigma) = (p\sigma)(p'\sigma) = 1$$

as required.

Conversely, $q\sigma$ is the unique right inverse of $q'\sigma$, so if

$$(q'u_1)\sigma = (q'\sigma)(u_1\sigma) = 1$$

then we must have $u_1\sigma = q\sigma$. Similarly, if $(u_2 p')\sigma = 1$ then $u_2\sigma = p\sigma$, and so we deduce that

$$u\sigma = (u_1 u_2)\sigma = (qp)\sigma$$

as required. □

5.2 Rational polycyclic monoid automata

We are now ready to prove our main theorem about M-automata with rational target sets where M is a polycyclic monoid.

Theorem 5.2.1. *Suppose $L \in F_{Rat}(P^0(X))$. Then L is a finite union of languages, each of which is the concatenation of one or two languages in $F_1(P^0(X))$.*

Proof. Let $M = P^0(X)$ and let A be an M-automaton with rational target set R accepting the language L. By Corollary 5.1.5 there exists an integer n and regular languages $Q_1, \ldots, Q_n \subseteq Q_X^*$ and $P_1, \ldots P_n \subseteq P_X^*$ such that

$$R = R_0 \cup \bigcup_{i=1}^{n} (Q_i P_i)\sigma.$$

where either $R_0 = \emptyset$ or $R_0 = \{0\}$ depending on whether $0 \in R$. For $1 \leq i \leq n$, we let $R_i = (Q_i P_i)\sigma$. We wish to split the language L into a union of languages in a similar way. Consider the rational subset R_i for some $i \in \{0, \ldots, n\}$.

For $i = 0, \ldots, n$ let A_i be the M-automaton with the same states and edges as A, but with rational target set R_i. Letting L_i be the language accepted by A_i, we see that

$$L = L_0 \cup L_1 \cup \cdots \cup L_n.$$

Clearly it suffices to show that each L_i is a finite union of languages, each of which is the concatenation of at most two languages in $F_1(M)$.

We begin with L_0. If $R_0 = \emptyset$ then $L_0 = \emptyset$, so assume that $R_0 = \{0\}$. Let $Z = \{u \in \Sigma_X^* \mid u\sigma = 0\}$ and $W = \{w \in \Sigma_X^* \mid w\sigma = 1\}$. By considering the rewriting system Λ from the proof of Theorem 5.1.1 we note that the only rule which has z on the right hand side but does not contain a z on the left hand side is the rule: $p_x q_y \to z$. The other rules with z on the right tell us that any expression containing z will eventually reduce to just z. So we conclude that $u \in Z$ if and only if either u contains the letter z, or u factorizes as $u_1 p_x u_2 q_y u_3$ where $x, y \in X$, $x \neq y$ and $u_1, u_2, u_3 \in \Sigma_X^*$ are such that u_2 represents the identity, that is, such that $u_2 \in W$.

Thus,

$$Z \;=\; \Sigma_X^*\ \{z\}\ \Sigma_X^*\ \cup \bigcup_{x,y\in X, x\neq y} \Sigma_X^*\ \{p_x\}\ W\ \{q_y\}\ \Sigma_X^*.$$

Let $\psi_1 = \{\epsilon\} \times \Sigma_X^*$ and $\psi_2 = \{(w,w) \mid w \in \Sigma_X^*\}$. Clearly $\psi_1, \psi_2 \subseteq \Sigma_X^* \times \Sigma_X^*$ are rational, so

$$\psi = \psi_1(\epsilon, z)\psi_1 \cup \bigcup_{x,y\in X, x\neq y} \psi_1(\epsilon, p_x)\psi_2(\epsilon, q_y)\psi_1$$

is also rational. We claim that $Z = \psi(W)$. Indeed, suppose $(u,v) \in \psi$ for some $u \in W$. Then either $u = \epsilon$ and v contains the letter z so that $v \in Z$ or there exists some $x, y \in X$ with $x \neq y$ such that $(u,v) \in \psi_1(\epsilon, p_x)\psi_2(\epsilon, q_y)\psi_1$. In the latter case we must have $(u,v) = (w, v_1 p_x w q_y v_2)$ for some $v_1, v_2 \in \Sigma_X^*$. But since $w = u \in W$ we have $v = v_1 p_x u q_y v_2 = v_1 p_x q_y v_2$ and so $v \in Z$ as required.

Conversely, assume that $v \in Z$. Then either v is equal to z or $v = v_1 p_x w q_y v_2$ for some $v_1, v_2 \in \Sigma_X^*$, $x \neq y$ and $w \in W$. In the former case $(\epsilon, v) \in \psi_1(\epsilon, z)\psi_1 \subseteq \psi$ which since $\epsilon \in W$ means that $v \in \psi(W)$. In the latter case $(w, v) \in \psi_1(\epsilon, p_x)w(\epsilon, q_y)\psi_1 \subseteq \psi$ and so v is again in $\psi(W)$ as required. This proves that the set Z is a rational transduction of W.

By Proposition 4.3.4, L_0 is a rational transduction of the language Z. Since the class of rational transductions is closed under composition (Theorem 2.2.10), it follows that L is a rational transduction of W, and hence by Proposition 3.0.2 that $L_0 \in F_1(M)$, as required.

We now turn our attention to the languages L_i for $i \geq 1$. Recall that L_i is accepted by an M-automaton with target set $R_i = (Q_i P_i)\sigma$. Let

$$P_i' \;=\; \{(p', \epsilon) \mid p \in P_i\} \;\subseteq\; Q_X^* \times \Sigma^*$$

and similarly

$$Q_i' \;=\; \{(q', \epsilon) \mid q \in Q_i\} \;\subseteq\; P_X^* \times \Sigma^*.$$

It is clear that the languages P_i' and Q_i' are rational. Indeed, given a finite automaton accepting P_i one may construct an automaton accepting P_i' in the same manner as in the proof of Proposition 2.2.6. Let A_P and A_Q be finite automata accepting P_i'

and Q'_i respectively, and assume without loss of generality that the first component of every edge label is either a single letter in Σ_X or the empty word ϵ.

By Proposition 4.3.4 there is a rational transduction $\rho \subseteq \Sigma_X^* \times \Sigma^*$ (depending on i) such that $w \in L_i$ if and only if $(u, w) \in \rho$ for some $u \in \Sigma_X^*$ such that $u\sigma \in R_i$. Let A be an automaton recognizing ρ, again with the property that the first component of every edge label is either a single letter in Σ_X or the empty word ϵ. We construct a new automaton B with

- vertex set the disjoint union of the state sets of A_Q, A, and A_P;
- all the edges of A_Q, A and A_P;
- initial vertex the initial vertex of A_Q;
- terminal vertices the terminal vertices of A_P;
- an extra edge, labelled (ϵ, ϵ), from each terminal vertex of A_Q to the initial vertex of A and
- an extra edge labelled (ϵ, ϵ), from each terminal vertex of A to the initial vertex of A_P.

It is immediate that B recognizes the relation

$$\tau \;=\; Q'_i \rho P'_i \;=\; \{(q'xp', w) \mid q \in Q_i, p \in P_i, (x, w) \in \rho\} \;\subseteq\; \Sigma_X^* \times \Sigma^*$$

and again has the property that the first component of every edge label is either a single letter or the empty word.

Let Q be the vertex set of A, viewed as a subset of the vertex set of B. For each vertex $y \in Q$, we let K_y be the language of all words w such that (u, w) labels a path in B from the initial vertex of B to y for some u with $u\sigma = 1$. By considering B as a transducer but with terminal vertex y, we see that K_y is a rational transduction of the identity language of $P(X)$, and hence by Proposition 4.3.4 lies in the class $F_1(P(X))$.

Dually, we let L_y be the language of all words w such that (u, w) labels a path in B from y to a terminal vertex for some u with $u\sigma = 1$. This time by considering B as

a transducer but with initial vertex y, we see that L_y is also a rational transduction of the identity language of $P(X)$, and hence also lies in $F_1(P(X))$.

We claim that

$$L_i = \bigcup_{y \in Q} K_y L_y,$$

which will clearly suffice to complete the proof.

Suppose first that $w \in L_i$. Then there exists a word $u \in \Sigma_X^*$ such that $u\sigma \in R_i$ and that $(u, w) \in \rho$. Since $R_i = (Q_i P_i)\sigma$ we have $u\sigma = (qp)\sigma$ for some $q \in Q_i$ and $p \in P_i$. Note that $(q'up', w) \in \tau$ is accepted by B. By Proposition 5.1.6, u admits a factorization $u = u_1 u_2$ such that $(q'u_1)\sigma = 1$ and $(u_2 p')\sigma = 1$. Now in view of our assumption on the edge labels of B, w must admit a factorization $w = w_1 w_2$ such that B has a path from the initial vertex to some vertex y labelled $(q'u_1, w_1)$ and a path from y to a terminal vertex labelled $(u_2 p', w_2)$; moreover, the vertex y can clearly be assumed to lie in Q. Since $(q'u_1)\sigma = 1 = (u_2 p')\sigma$, it follows that $w_1 \in K_y$ and $w_2 \in L_y$ so that $w = w_1 w_2 \in K_y L_y$, as required.

Conversely, suppose $y \in Q$ and that $w = w_1 w_2$ where $w_1 \in K_y$ and $w_2 \in L_y$. Then B has a path from the initial vertex to vertex y labelled (u_1, w_1) and a path from the vertex y to a terminal vertex labelled (u_2, w_2) for some u_1 and u_2 with $u_1\sigma = u_2\sigma = 1$. Since $y \in Q$, it follows from the definition of B that $u_1 = q'v_1$ and $u_2 = v_2 p'$ for some $q \in Q_i$ and $p \in P_i$ and v_1 and v_2 such that $(v_1 v_2, w) \in \rho$. But now

$$(q'v_1)\sigma = u_1\sigma = 1,$$

and

$$(v_2 p')\sigma = u_2\sigma = 1,$$

so we deduce by Proposition 5.1.6 that $v_1\sigma = q\sigma$ and $v_2\sigma = p\sigma$. But then

$$(v_1 v_2)\sigma = (qp)\sigma \in R_i \subseteq R$$

and $(v_1 v_2, w) \in \rho$, from which it follows that $w \in L_i$ as required.

Thus, we have written L as a finite union of languages L_i where each L_i either lies in $F_1(M)$ (in the case $i = 0$) or is a finite union of concatenations of two languages in $F_1(M)$. This completes the proof. □

In the case that $|X| \geq 2$, we have $P^0(X) = P(X)$ and $F_1(P(X))$ is the class of context-free languages, which is closed under both finite union and concatenation. Hence, we obtain the following easy consequence.

Theorem 5.2.2. *If $|X| \geq 2$ then $F_{Rat}(P(X))$ is the class of context-free languages.*

Proof. By Theorem 5.2.1 the family of languages $F_{Rat}(P(X))$ for $|X| = 2$ contains only languages which are a finite union of concatenations of one or two languages from $F_1(P(X)$. By Theorem 3.3.1 $F_1(P(X))$ is exactly the family of context free languages. Since $F_1(P(X))$ is closed under union and concatenation (Proposition 5.1.3), $F_{Rat}(P(X)) \subseteq F_1(P(X))$. But from Proposition 4.2.2, $F_1(P(X)) \subseteq F_{Rat}(P(X))$, and the result follows. □

In the case $|X| = 1$, we have that $P^0(X)$ is isomorphic to the bicyclic monoid $B = P(X)$ with a zero adjoined. Combining Theorem 5.2.1 with Proposition 4.1.3 and Theorem 4.4.1 we thus obtain:

Corollary 5.2.3. *Every language in $F_{Rat}(B)$ is a finite union of languages, each of which is the concatenation of one or two partially blind one-counter languages.*

Proof. From Theorem 5.2.1 we may conclude that languages in $F_{Rat}(B)$ take the form of a finite union of languages, each of which is a concatenation of one or two languages from $F_1(B^0)$. By Proposition 4.1.3 $F_1(B^0) = F_1(B)$ and hence each language in the union is a concatenation of one or two languages from $F_1(B)$. $F_1(B)$ is exactly the family of partially blind one-counter languages, and the result follows. □

Since the class $F_1(B)$ of partially blind one-counter languages is not closed under concatenation, we cannot conclude that $F_{Rat}(B) = F_1(B)$. Indeed, the following result shows that this is not the case.

Theorem 5.2.4. *The language*

$$\{a^i b^i a^j b^j \mid i, j \geq 0\} \subseteq \{a, b\}^*$$

lies in $F_{Rat}(B)$ *but not in* $F_1(B)$.

Proof. Let $L = \{a^i b^i a^j b^j \mid i, j \geq 0\}$. First, we claim that the B-automaton with rational target set shown in Figure 5.1 accepts the language L. Indeed, it is easily seen to accept exactly pairs of the form

$$(p^{i_0} q^{i_1} qpp^{i_2} q^{i_3}, a^{i_0} b^{i_1} a^{i_2} b^{i_3}) = (p^{i_0} q^{i_1+1} p^{i_2+1} q^{i_3}, a^{i_0} b^{i_1} a^{i_2} b^{i_3})$$

for $i_0, i_1, i_2, i_n \in \mathbb{N}$.

First assume that $i_0 > i_1$. Then using the fact that $pq = 1$

$$p^{i_0} q^{i_1+1} p^{i_2+1} q^{i_3} = p^j p^{i_2+1} q^{i_3} \neq qp$$

for $j \geq 0$. If $i_0 < i_1$ then

$$p^{i_0} q^{i_1+1} p^{i_2+1} q^{i_3} = q^j p^{i_2+1} q^{i_3} \neq qp$$

with $j > 1$. So we conclude that $i_0 = i_1$ if $p^{i_0} q^{i_1+1} p^{i_2+1} q^{i_3} = qp$. Next, assume that $i_2 > i_3$. Then

$$p^{i_0} q^{i_1+1} p^{i_2+1} q^{i_3} = qp^j \neq qp$$

as $j > 1$. If $i_2 < i_3$ then

$$p^{i_0} q^{i_1+1} p^{i_2+1} q^{i_3} = q^{j+1} \neq qp,$$

$j > 1$. On the other hand if $i_0 = i_1$ and $i_2 = i_3$ then $p^{i_0} q^{i_1+1} p^{i_2+1} q^{i_3} = qp$. This suffices to establish the claim that $L \in F_{Rat}(B)$.

Assume now for a contradiction that $L \in F_1(B)$. Then there exists a B-automaton A accepting L, with N vertices say. For $i \geq 0$ let π_i be an accepting path for $a^i b^i a^i b^i$. Suppose without loss of generality that the right-hand sides of edge labels in A are all a, b or ϵ. Then we can write $\pi_i = \alpha_i \beta_i \gamma_i \delta_i$ and where α_i has label (s_i, a^i), β_i has label (t_i, b^i), γ_i has label (u_i, a^i) and δ_i has label (v_i, b^i) for some $s_i, t_i, u_i, v_i \in B$.

The proof will proceed by considering loops (that is, closed paths) in the automaton A; we begin by introducing some terminology to describe particular types of loops. A loop with label $(q^k p^j, x)$ is called an *increment loop* if $j > k$, a *stable loop* if $k = j$ and a *decrement loop* if $k > j$. We call the loop an *epsilon loop* if $x = \epsilon$ and a *non-epsilon* loop otherwise. A path which does not traverse any loops is called a *simple path*.

First notice that since there are only finitely many simple paths, there exists a constant K such that every simple path in A has label of the form $(q^g p^h, x)$ with $g + h < K$.

Consider paths of the form α_i. We claim that for all but at most KN values of i, the path α_i contains a non-epsilon increment loop. For all $i \geq N$, we can write $\alpha_i = \alpha_i^{(1)} \alpha_i^{(2)}$ where $\alpha_i^{(1)}$ has label $(s_i^{(1)}, a^{i-N})$ and $\alpha_i^{(2)}$ has label $(s_i^{(2)}, a^N)$.

Note that the only elements of B which generate a right ideal [left ideal] including the identity element, are those of the form p^k [respectively q^k] for some $k \geq 0$. Thus, we must have that both s_i and $s_i^{(1)}$ are powers of p, and that v_i is a power of q. In particular, we can let $f_i \geq 0$ be such that $s_i^{(1)} = p^{f_i}$.

Next suppose i is such that $\alpha_i^{(1)}$ does not traverse an increment loop. Let α_i' be the path obtained from $\alpha_i^{(1)}$ by removing all loops, and suppose α_i' has label $(q^g p^h, a^l)$. Since none of the loops removed were increment loops, it follows easily that

$$f_i \ \leq \ h - g \ \leq \ h + g \ \leq \ K.$$

Suppose now for a contradiction than more than KN values of $i \geq N$ are such that α_i' contains no increment loop. Then by the pigeonhole principle, there exist $i \neq j$

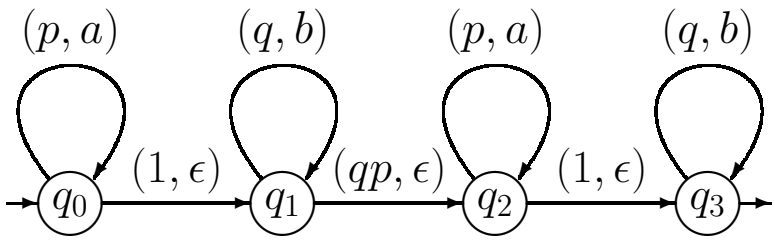

Figure 5.1: A rational B-automaton with target set $\{qp\}$, accepting the language $\{a^i b^i a^j b^j \mid i, j \geq 0\}$.

with $i \geq N$ and $j \geq N$ such that $f_i = f_j$ and the paths $\alpha_i^{(1)}$ and $\alpha_j^{(1)}$ end at the same state. But now the composition $\alpha_i^{(1)}\alpha_j^{(2)}\beta_j\gamma_j\delta_j$ is an accepting path with label

$$\begin{aligned}(s_i^{(1)}s_j^{(2)}t_ju_jv_j, a^{i-N}a^Nb^ja^jb^j) &= (p^{f_i}s_j^{(2)}t_ju_jv_j, a^ib^ja^jb^j)\\ &= (s_j^{(1)}s_j^{(2)}t_ju_jv_j, a^ib^ja^jb^j)\\ &= (s_jt_ju_jv_j, a^ib^ja^jb^j)\\ &= (1, a^ib^ja^jb^j)\end{aligned}$$

so that $a^ib^ja^jb^j$ is accepted by A, giving a contradiction. Thus, we have established that for all but KN values of $i \geq N$, the path $\alpha_i^{(1)}$ must traverse an increment loop. Hence, for all but $KN + N = (K+1)N$ values of $i \geq 0$, the path $\alpha_i^{(1)}$ must traverse an increment loop.

Now let i be such that $\alpha_i^{(1)}$ traverses an increment loop and suppose for a contradiction that α_i does not traverse a non-epsilon increment loop. Consider the path $\alpha_i^{(2)}$. Clearly, since this path has label with right-hand-side a^N, and the right-hand-sides of edge labels in the automaton are single letters or ϵ, this path must traverse a non-epsilon loop. Since α_i does not traverse a non-epsilon increment loop, $\alpha_i^{(2)}$ must traverse a non-epsilon stable or decrement loop, say with label (q^gp^h, a^k) where $0 \leq h \leq g$ and $0 < k$. We also know that $\alpha_i^{(1)}$ traverses an epsilon increment loop, say with label (q^xp^y, ϵ) where $0 \leq x < y$. Clearly, by traversing the latter loop an additional $(g-h)$ times and the former loop an additional $(y-x)$ times, we obtain an accepting path for the word $a^{i+(y-x)k}b^ia^ib^i$, which gives the required contradiction.

Thus, we have shown that for all but at most $(K+1)N$ values of i, the path α_i traverses a non-epsilon increment loop. A left-right symmetric argument can be used to establish firstly that each $v_i = q^{g_i}$ for some $g_i \geq 0$, and then that for i sufficiently large, δ_i must traverse a non-epsilon decrement loop. Thus, for all but at most $2(K+1)N$ values of i, the paths α_i and δ_i traverse respectively a non-epsilon increment loop and a non-epsilon decrement loop.

Now choose i such that this holds, and let (q^jp^k, a^m) be the label of the subpath of π_i consisting of traversals of a non-epsilon increment loop in α_i and let $(q^{j'}p^{k'}, b^{m'})$ similarly label the subpath consisting of traversals of a non-epsilon decrement loop

in δ_i where $k > j$, $k' > j'$ and $m, m' > 0$. Let π_i' be the path obtained from π_i by traversing the given increment loop path an additional $j' - k'$ times, and the given decrement loop path an additional $k - j$ times. Then π_i has label of the form

$$\left(t(q^j p^k)^{(j'-k')+1} u (q^{j'} p^{k'})^{(k-j)+1} v,\ a^{i+m(j'-k')} b^i a^i b^{i+m'(k-j)} \right)$$

where t, u and v are such that π has label

$$\left(tq^j p^k u q^{j'} p^{k'} v,\ a^i b^i a^i b^i \right)$$

so that in particular $tq^j p^k u q^{j'} p^{k'} v = 1$. Now by our argument above regarding right and left ideals, the element $tq^j \in B$ must be a power of p, while $q^{j'} v \in B$ must be a power of q. Noting that powers of p commute with each other, and powers of q commute with each other, we get

$$\begin{aligned} t(q^j p^k)^{(j'-k')+1} u (q^{j'} p^{k'})^{(k-j)+1} v &= tq^j p^{(k-j)(j'-k')} p^k u q^{j'} q^{(k-j)(j'-k')} p^{k'} v \\ &= p^{(k-j)(j'-k')} tq^j p^k u q^{j'} p^{k'} v q^{(k-j)(j'-k')} \\ &= p^{(k-j)(j'-k')} 1 q^{(k-j)(j'-k')} \\ &= 1. \end{aligned}$$

Therefore π_i' is an accepting path. Thus, the automaton accepts the word

$$a^{i+m(j'-k')} b^i a^i b^{i+m'(k-j)}$$

which is not in the language L, giving the required contradiction. This completes the proof that $L \notin F_1(B)$. □

It is possible, however, to describe concatenations of partially blind one-counter languages using partially blind two-counter automata. Indeed more generally we have the following proposition.

Proposition 5.2.5. *Let M_1 and M_2 be monoids and L_1 and L_2 languages over the same alphabet. If $L_1 \in F_1(M_1)$ and $L_2 \in F_1(M_2)$ then $L_1 L_2 \in F_1(M_1 \times M_2)$.*

Proof. By Proposition 4.3.4 for $i = 1, 2$ there are alphabets Ω_i, morphisms $\omega_i : \Omega_i^* \to M_i$ and rational transductions $\rho_i \subseteq \Omega_i^* \times \Sigma^*$ such that $L_i = \{1\}\omega_i^{-1}\rho_i$. Assume

without loss of generality that Ω_1 and Ω_2 are disjoint, and let $\Omega = \Omega_1 \cup \Omega_2$. Then there is a natural morphism $\omega : \Omega^* \to M_1 \times M_2$ extending ω_1, ω_2. Now let ρ be the *product* of ρ_1 and ρ_2:

$$\rho \;=\; \{(u_1u_2, w_1w_2, \mid (u_1, w_1) \in \rho_1, (u_2, w_2) \in \rho_2)\} \;\subseteq\; \Omega^* \times \Sigma^*.$$

Then ρ is a rational transduction from Ω^* to Σ^*. Clearly, if $u_1 \in \Omega_1^*$ and $u_2 \in \Omega_2^*$ then u_1u_2 represents the identity element in $M_1 \times M_2$ if and only if u_1 and u_2 represent the identity elements in M_1 and M_2 respectively. It follows that w is in the image under ρ of the identity language of $M_1 \times M_2$ if and only if $w = w_1w_2$ where $w_1 \in L_1$ and $w_2 \in L_2$, so that $w \in L_1L_2$. Thus, L_1L_2 is a rational transduction of the identity language of $M_1 \times M_2$, so applying Proposition 3.0.2 we see that $L_1L_2 \in F_1(M_1 \times M_2)$ as required. □

Corollary 5.2.6. *$F_{Rat}(B) \subseteq F_1(B^2) \cap CFL$.*

Proof. Since classes of the form $F_1(M)$ are closed under union, Theorem 5.2.1 and Proposition 5.2.5 combine to give the inclusion of $F_{Rat}(B)$ in $F_1(B^2)$. Also since B is a submonoid of $P(X)$ with $|X| = 2$, $F_{Rat}(B) \subseteq F_{Rat}(P(X)) = CFL$ by Theorem 5.2.2. □

Chapter 6

Completely simple semigroups

In this chapter we consider language classes $F_{Rat}(S)$ for semigroups S taken from the important classes of completely simple and completely 0-simple semigroups. Some of the material in this chapter has been published in [51].

Given the context of our study, the property of being finitely generated is of much importance. The following results from [2] precisely characterise the conditions under which a given Rees matrix semigroup may be finitely generated.

Theorem 6.0.7. *Let T be a semigroup, let I and J be index sets, let $P = (P_{ji})_{j\in J, i\in I}$ be a $J \times I$ matrix with entries from T, and let U be the ideal of T generated by the set $\{P_{ji} \mid j \in J, i \in I\}$ of all entries of P. Then the Rees matrix semigroup $M(T; I, J; P)$ is finitely generated if and only if the following three conditions are satisfied:*

(i) *both I and J are finite;*

(ii) *T is finitely generated;*

(iii) *the set $T \setminus U$ is finite.*

Let $M^0(T; I, J; P)$ be a Rees matrix semigroup with T a semigroup with zero. By noting the fact that we may construct a Rees matrix semigroup with zero by taking the Rees quotient of $M(T^0; I, J; P)$ with respect to the ideal $I \times \{0\} \times J$, and that if the index sets I and J are finite then necessarily the ideal is finite, we conclude the following.

Corollary 6.0.8. *Let T be a semigroup with zero, let I and J be index sets, and let $P = (P_{ji})_{j\in J, i\in I}$ be a $J \times I$ matrix with entries from T, and let U be the ideal of T generated by the set $\{p_{ji} \mid j \in J, i \in I\}$ of all entries of P. Then the Rees matrix semigroup with zero $M^0(T; I, J; P)$ is finitely generated if and only if the following three conditions are satisfied:*

(i) *both I and J are finite;*

(ii) *T is finitely generated;*

(iii) *the set $T \setminus U$ is finite.*

6.1 Rational subsets

In this section we consider the Rees matrix construction and how this affects the structure of the rational subsets of general Rees matrix semigroups and completely simple semigroups.

We require the following proposition.

Proposition 6.1.1. *Let $S = M^0(T; I, J; P)$ be a Rees matrix semigroup with zero over a semigroup T, and let $X \subseteq S$ be a rational subset. Then the set*

$$X_{ij} = \{g \in T \mid (i, g, j) \in X\}$$

is a rational subset of T.

Proof. Let A be a finite automaton over S accepting the rational subset X with state set Q. Let J' be the set of all $j' \in J$ which appear in edge labels of A; note that J' is necessarily finite. We construct from A a new finite automaton B over T with

- state set $(Q \times J') \cup \{q'_0\}$ where q'_0 is a new symbol;
- start state q'_0;
- terminal states (q, j) such that q is a terminal state of A;

- an edge from q_0' to (q_1, j_1) labelled t_1 whenever A has an edge from the initial state to q_1 labelled (i, t_1, j_1);
- for every $j_1 \in J'$, an edge from (q_1, j_1) to (q_2, j_2) labelled $P_{j_1 i_2} t_2$ whenever A has an edge from q_1 to q_2 labelled (i_2, t_2, j_2) with $P_{j_1 i_2} \neq 0$.

Since J' is finite and A has finitely many states and edges, we deduce that B has finitely many states and edges. Next we must show that the subset accepted by B is exactly X_{ij}. Let $t \in X_{ij}$. Then $(i, t, j) \in X$ labels a path in A from the initial state to some terminal state. Clearly this path cannot contain edges labelled 0, so it must have the form

$$\mathbf{p_0} \xrightarrow{(i_1,t_1,j_1)} \mathbf{p_1} \xrightarrow{(i_2,t_2,j_2)} \mathbf{p_2} \xrightarrow{(i_3,t_3,j_3)} \cdots \xrightarrow{(i_{m-1},t_{m-1},j_{m-1})} \mathbf{p_{m-1}} \xrightarrow{(i_m,t_m,j_m)} \mathbf{p_m}$$

where p_0 is the initial state of A and p_m is a terminal state. Since the path is labelled (i, t, j) we have

$$(i, t, j) \;=\; (i_1, t_1, j_1)(i_2, t_2, j_2) \ldots (i_m, t_m, j_m)$$

so that $i_1 = i$, $j_m = j$. Now it follows easily from the construction of B that there exists a path

$$\mathbf{q_0'} \xrightarrow{t_1} (\mathbf{p_1}, \mathbf{j_1}) \xrightarrow{P_{j_1 i_2} t_2} (\mathbf{p_2}, \mathbf{j_2}) \ldots (\mathbf{p_{m-1}}, \mathbf{j_{m-1}}) \xrightarrow{P_{j_{m-1} i_m} t_m} (\mathbf{p_m}, \mathbf{j}),$$

where (p_m, j) is a terminal state of B, so that B accepts

$$t = t_1 P_{j_1 i_2} t_2 P_{j_2 i_3} \ldots P_{j_{m-1} i_m} t_m.$$

Thus $X_{ij} \subseteq L(B)$.

Conversely, assume that $t \in T$ is accepted by B. Then there exists a path through B from the initial state to some terminal state labelled with t. It follows from the definition of B that this path must have the form

$$\mathbf{q_0'} \xrightarrow{t_1} (\mathbf{p_1}, \mathbf{j_1}) \xrightarrow{P_{j_1 i_2} t_2} (\mathbf{p_2}, \mathbf{j_2}) \ldots (\mathbf{p_{m-1}}, \mathbf{j_{m-1}}) \xrightarrow{P_{j_{m-1} i_m} t_m} (\mathbf{p_m}, \mathbf{j}),$$

where p_m is a terminal state in A,

$$t \;=\; t_1 P_{j_1 i_2} t_2 P_{j_2 i_3} t_3 \ldots P_{j_{m-1} i_m} t_m$$

and A has a path

$$\mathbf{p_0} \xrightarrow{(i,t_1,j_1)} \mathbf{p_1} \xrightarrow{(i_2,t_2,j_2)} \mathbf{p_2} \xrightarrow{(i_3,t_3,j_3)} \cdots \xrightarrow{(i_{m-1},t_{m-1},j_{m-1})} \mathbf{p_{m-1}} \xrightarrow{(i_m,t_m,j)} \mathbf{p_m}$$

where p_0 is the initial state of A. Hence, A accepts the element

$$\begin{aligned}(i,t_1,j_1)(i_2,t_2,j_2)\ldots(i_m,t_m,j) &= (i,t_1P_{j_1i_2}t_2P_{j_2i_3}t_3\ldots P_{j_{m-1}i_m}t_m,j)\\ &= (i,t,j).\end{aligned}$$

So $(i,t,j) \in X$ and hence $t \in X_{ij}$.

So the automaton B accepts exactly the set X_{ij}, and hence X_{ij} is a rational subset of T.

□

As a corollary, we obtain a result about the intersections of rational subsets with maximal subgroups in completely simple semigroups.

Corollary 6.1.2. *Let H be a maximal subgroup of a completely simple or completely 0-simple semigroup S. Let X be a rational subset of S. Then $X \cap H$ is a rational subset of H.*

Proof. By the Rees theorem, we may assume that S is a Rees matrix semigroup without zero - $M(G;I,J;P)$, or a regular Rees matrix semigroup with zero - $S = M^0(G;I,J;P)$, over a group G. It follows easily from the definition of the Rees matrix construction that either $H = \{0\}$ or

$$H \;=\; \{(i,g,j) \mid g \in G\}$$

for some $i \in I$ and $j \in J$ with $P_{ji} \neq 0$. In the former case the result is trivial, so we assume the latter. By Proposition 6.1.1, the set

$$X_{ij} \;=\; \{g \in G \mid (i,g,j) \in X\} \;=\; \{g \in G \mid (i,g,j) \in H \cap X\}$$

is a rational subset of G. It follows that

$$P_{ji}X_{ij} \;=\; \{P_{ji}g \mid g \in X_{ij}\} \;=\; \{P_{ji}g \mid (i,g,j) \in X\}$$

is also a rational subset of G. Now define a map

$$\phi \ : \ G \to H, \ g \mapsto (i, P_{ji}^{-1} g, j)$$

where P_{ji}^{-1} is the inverse of P_{ji} in the group G. It is readily verified that ϕ is an isomorphism from G to H, and so the image

$$\begin{aligned}(P_{ji}X_{ij})\phi &= \{(i, P_{ji}^{-1}g, j) \mid g \in P_{ji}X_{ij}\} \\ &= \{(i, P_{ji}^{-1}P_{ji}g, j) \mid (i, g, j) \in X\} \\ &= \{(i, g, j) \mid (i, g, j) \in X\} \\ &= X \cap H\end{aligned}$$

is a rational subset of H, as required. □

In a completely simple semigroup, where every element lies in a maximal subgroup, Corollary 6.1.2 easily yields the following complete characterisation of rational subsets.

Theorem 6.1.3. *The rational subsets of a completely simple semigroup are exactly the finite unions of rational subsets of maximal subgroups.*

Proof. Let S be a completely simple semigroup. If $X_1, \ldots, X_n$ are rational subsets of maximal subgroups of S then certainly they are rational subsets of S, and by Proposition 2.2.5 so is their union.

Conversely, suppose X is a rational subset of S. Then X is accepted by a finite automaton over S. Let $I' \subseteq I$ and $J' \subseteq J$ be the sets of indices appearing in edge labels of the automaton. Then let P' denote the $|I'| \times |J'|$ sandwich matrix consisting of only those rows and columns appearing in I' and J'. Similarly let $G' \subseteq G$ be the subgroup generated by elements $g \in G$ appearing in edge labels in the automaton and elements P'_{ji} appearing in the sandwich matrix P'. Let $S' = M(G'; I', J'; P')$ be the Rees matrix semigroup constructed from these sets. Since the automaton is finite it is clear to see that the sets I' and J' are finite, and that G' is finitely generated. Similarly the matrix P' cannot contain any zero entries and so by the Rees theorem S' is completely simple. We also note that the ideal of G' generated by elements appearing

in P' is precisely the group G' and hence by Theorem 6.0.7 we see that S' is finitely generated. So X lies inside a finitely generated completely simple subsemigroup S' of S. Now S' is the union of finitely many maximal subgroups, so X is the union of its intersections with these subgroups. By Corollary 6.1.2 these intersections are rational, so X is a finite union of rational subsets of maximal subgroups of S'. But maximal subgroups of S' are subgroups of S, and hence lie in maximal subgroups of S'. It follows that X is a finite union of rational subsets of maximal subgroups of S, as required. □

Proposition 6.1.4. *Let $S = M(T; I, J; P)$ or $S = M^0(T; I, J; P)$ be a Rees matrix semigroup with or without zero over a semigroup T, and let $P' \subseteq T$ be the set of non-zero entries of the sandwich matrix P. Suppose $T = P'T$ or $T = TP'$. Then for any $i \in I$, $j \in J$ and rational subset X of T, the set*

$$\{(i, t, j) \mid t \in X\}$$

is a rational subset of S.

Proof. By symmetry of assumption, it suffices to consider the case in which $T = P'T$. Let A be a finite automaton over T accepting X, with state set Q. Let $Y \subseteq T$ be the set of edge labels in A, and for every $t \in Y$, choose $j_t \in J$, $i_t \in I$ and $s_t \in T$ such that $t = P_{j_t i_t} s_t$. Let

$$J' = \{j_t \mid t \in Y\} \cup \{j\}.$$

Then J' is a finite subset of J. We define a new automaton B over S with

- state set $(Q \times J') \cup \{q_0\}$ where q_0 is a new symbol;
- initial state q_0;
- terminal states (q, j) such that q is a terminal state of A;
- for every edge in A from the start state to a state q labelled t, and every $j' \in J'$, an edge from q_0 to (q, j') labelled (i, t, j');
- for every edge in A from a state p to a state q labelled t, and every $j' \in J'$, an edge from (p, j_t) to (q, j') labelled (i_t, s_t, j').

We first note that since J' and Q are finite, the state set of the new automaton B is also finite. Let $x \in X$. Then there exists a path through A connecting the initial state to a terminal state labelled by x. Assume that the accepting path is labelled by $x = x_1 x_2 \dots x_n$ for $x_i \in Y$ for $i = 1, \dots, n$, with the x_i not necessarily distinct. By construction, for every edge from state p to state q in A labelled by t there exists an edge labelled by (i_t, s_t, j') from state (p, j_t) to state (q, j') for every $j' \in J'$. If we set $j_k = j_{x_{k+1}}$ for $k = 1, \dots, n-1$, and set $j_n = j$ then it follows that B has an accepting path labelled

$$\begin{aligned}(i, x_1, j_{x_2})(i_{x_2}, s_{x_2}, j_{x_3}) \dots (i_{x_n}, s_{x_n}, j) &= (i, x_1 P_{j_{x_2} i_{x_2}} s_{x_2} \dots s_{x_{n-1}} P_{j_{x_n} i_{x_n}} s_{x_n}, j) \\ &= (i, x_1 x_2 \dots x_n, j) \\ &= (i, x, j)\end{aligned}$$

and so (i, x, j) is accepted by B and $i \times L(A) \times j \subseteq L(B)$.

For the other direction, let $(i, x, j) \in L(B)$. Then there exists a path connecting the initial state to some terminal state labelled by (i, x, j). It follows from the definition of B that the path has label

$$(i, x_1, j_2)(i_2, s_2, j_3) \dots (i_n, s_n, j)$$

for $x_1, s_2, \dots s_n \in Y$ (not necessarily distinct) and $j_2, \dots j_n \in J'$ (again, not necessarily distinct). Since the path has label (i, x, j) we have that

$$x_1 P_{j_2 i_2} s_2 \dots s_{n-1} P_{j_n i_n} s_n = x.$$

Let $x_k = P_{j_k i_k} s_k$ for $k = 2, \dots, n$, then $x = x_1 \dots x_n$. So, reversing the construction, there exists a path through A labelled by $x_1 x_2 \dots x_n$ as required. Thus the set $\{(i, x, j) \mid x \in X\}$ is accepted by the automaton B. □

Note in particular that the conditions on the sandwich matrix in the hypothesis of Proposition 6.1.4 are satisfied in the case of a regular Rees matrix construction over a group.

As a corollary we obtain a result about the decidability of the rational subset problem for completely simple and completely 0-simple semigroups.

Corollary 6.1.5. *Let $S = M(T; I, J; P)$ or $S = M^0(T; I, J; P)$ be a finitely generated Rees matrix semigroup with or without zero over a semigroup T. If T has decidable rational subset problem then S has decidable rational subset problem.*

Proof. We prove the statement for Rees matrix semigroups with zero. Since any Rees matrix semigroup S without zero may be embedded into a Rees matrix semigroup S' with zero with the same maximal non-zero subgroup, and $F_{Rat}(S) \subseteq F_{Rat}(S')$, the result follows easily for Rees matrix semigroups without zero. It may also be proven directly by a similar method to below.

Let $\omega : \Omega^* \to T$ and $\sigma : \Sigma^* \to S$ be finite choices of generators for T and S respectively. For every $x \in \Sigma$ such that $x\sigma \neq 0$, suppose $x\sigma = (i_x, g_x, j_x)$ and let $w_x \in \Omega^*$ be a word with $w_x\omega = g_x$. For $j \in J$ and $i \in I$ such that $P_{ji} \neq 0$ let $w_{ji} \in \Omega^*$ be a word with $w_{ji}\omega = P_{ji}$.

Now suppose we are given a word $w = w_1 \dots w_n \in \Sigma^*$, where each $w_i \in \Sigma$, and a rational subset X of S. Clearly, we can test whether w represents 0 and, in the case that it does, whether $0 \in X$. Assume now that w does not represent 0. Then

$$w\sigma = (w_1\sigma) \dots (w_n\sigma) = (i_{w_1}, g_{w_1} P_{j_{w_1} i_{w_2}} g_{w_2} \dots g_{w_n}, j_{w_n}).$$

Let $Y = \{t \in T \mid (i_{w_1}, t, j_{w_n}) \in X\}$, so that $w\sigma \in X$ if and only if

$$(w_{g_{w_1}} w_{j_{w_1} i_{w_2}} w_{g_{w_2}} \dots w_{g_{w_n}})\omega = g_{w_1} P_{j_{w_1} i_{w_2}} g_{w_2} \dots g_{w_n} \in Y. \tag{6.1}$$

Now by Proposition 6.1.1, Y is rational and it follows moreover from the proof that we can effectively compute an automaton for Y. By assumption, we can solve the rational subset problem for Y, so we can decide whether (6.1) holds, as required. □

6.2 Rational semigroup automata

We now turn our attention to languages accepted by rational S-automata, where S is a Rees matrix semigroup. The first lemma will prove useful later.

Lemma 6.2.1. *Let $S = M^0(G; I, J; P)$ be a Rees matrix semigroup with zero and let $\omega : \Omega^+ \to S$ be a choice of generators for S. Then the set $\{z \in \Omega^+ \mid z\omega = 0\}$ is regular.*

Proof. For each $x \in \Omega$ such that $x\omega \neq 0$ suppose $x\omega = (i_x, g_x, j_x)$. Let $w \in \Omega^+$. Then $w\omega = 0$ if and only if either $w \in \Omega^* x \Omega^*$ where $x \in \Omega$ is such that $x\omega = 0$ or there exist two consecutive generators in w, x and y say, such that $P_{j_x i_y} = 0$.

In the former case the subset of generators $x \in \Omega$ such that $x\sigma = 0$ is necessarily finite since Ω is finite. Call this set Ω'.

In the latter case, the set of all possible pairs of generators $x, y \in \Omega$ such that $P_{j_x i_y} = 0$ is also finite. Call this set P'.

Then we may write the set $\{z \in \Omega^+ \mid z\omega = 0\}$ as

$$\Omega^* \Omega' \Omega^* \cup \Omega^* P' \Omega^*$$

so the set in question is regular. □

The next lemma simplifies the case of Rees matrix semigroups with zero, by allowing us to restrict attention to automata for which neither the initial set nor the terminal set contain zero.

Lemma 6.2.2. *Let $S = M^0(T; I, J; P)$ be a finitely generated Rees matrix semigroup with zero over a semigroup T, and suppose P contains a non-zero entry. If $L \in F_{Rat}(S)$ then L is accepted by an S-automaton with rational initial and terminal sets neither of which contain 0.*

Proof. Suppose L is accepted by an S-automaton A with rational initial set X_0 and rational terminal set X_1. Suppose first that $0 \in X_0$. If also $0 \in X_1$ then we have $0x \in X_1$ for all $x \in S$, so the language accepted is just the set of all words w such that (x, w) labels a path from the initial vertex to a terminal vertex of A for some $x \in S$. It follows that L is regular. We claim that L is accepted by an S-automaton with rational initial and terminal set $S \setminus \{0\}$. We note that by Lemma 6.2.1 and Proposition 4.2.4 this set is rational because S is rational. Indeed, let $(i, t, j) \in S$ such that $P_{ji} \neq 0$ for some $t \in T$. Then $(i, t, j)^n \neq 0$ for all $n \in \mathbb{N}$ and hence for an appropriate choice of generators $\sigma : \Omega^+ \to S$ there exist words of arbitrarily high length over Ω whose image under σ is contained in $S \setminus \{0\}$. So we may apply Lemma

4.4.2 to conclude that the regular language L is accepted by an S-automaton without zero in the initial or terminal sets.

On the other hand, if $0 \notin X_1$ then there is no $x \in S$ such that $0x \in X_1$; hence we may replace the initial set X_0 with $X_0 \setminus \{0\}$ without changing the language accepted. Indeed, we note that $\{0\}\sigma^{-1}$ is regular by Lemma 6.2.1 and hence by Proposition 4.2.4 $X_0 \setminus \{0\}$ is rational if X_0 is rational. Thus, we may assume that $0 \notin X_0$.

Clearly we can write $L = L_0 \cup L_1$ where L_1 is accepted by a S-automaton with initial set X_0 and terminal set $X_1 \setminus \{0\}$, and L_0 is accepted by an S-automaton with terminal set $\{0\}$ and rational initial set X_0. We show first that L_0 is regular.

Let $\sigma : \Omega^* \to S$ be a finite choice of generators for S. For each $x \in \Omega$ such that $x\sigma \neq 0$ suppose $x\sigma = (i_x, g_x, j_x)$. Now let $K \subseteq \Omega^*$ be the set of all words representing elements of the initial set of A, and let $K' \subseteq \Omega$ be the (necessarily finite) set of all final letters of words in K. It is easily seen that the language

$$\{v \in \Omega^* \mid (wv)\sigma = 0 \text{ for some } w \in K\}$$

is regular. Indeed, by a similar argument to Lemma 6.2.1 it consists of all words which

1. contain a generator representing zero; or
2. contain consecutive generators x and y with $P_{j_x i_y} = 0$; or
3. start with a generator y with $P_{j_x i_y} = 0$ for some $x \in K'$

and so can be easily described by a regular expression.

It now follows from Proposition 4.3.4 that L_0 is a rational transduction of a regular language and hence is itself regular. It follows that L is the union of L_1 with a regular language.

We next note that by combining Lemma 6.2.1 and Proposition 4.2.4 if X_1 is rational then $X_1 \setminus \{0\}$ is also rational. Now if $X_0^{-1}(X_1 \setminus \{0\})$ contains elements $w\sigma$ such that $w \in \Omega^+$ is of arbitrarily high length then by Lemma 4.4.2 the set of languages accepted by S-automata with initial set X_0 and terminal set $X_1 \setminus \{0\}$

contains the regular languages. It follows that both L_0 and L_1 are accepted by S-automata with initial set X_0 and terminal set $X_1 \setminus \{0\}$. Now just as in the proof of Theorem 4.4.5 we may construct an S-automaton with initial set X_0 and terminal set $X_1 \setminus \{0\}$ which accepts the union $L = L_0 \cup L_1$.

On the other hand if there exists an upper bound on the length of words $w \in \Omega^+$ such that $w\sigma \in X_0^{-1}(X_1 \setminus \{0\})$ then by Lemma 4.4.3 every language accepted by S-automata with initial set X_0 and terminal set $X_1 \setminus \{0\}$ is finite and hence regular. Therefore the language L_1 is itself regular, and the union $L_0 \cup L_1$ is regular. Now by Lemma 6.2.1 the set $S \setminus \{0\}$ is rational. We set $X_0' = X_1' = S \setminus \{0\}$, then by Lemma 4.4.2 there exists an S-automaton with initial set X_0' and terminal set $X_!'$ accepting the union $L_0 \cup L_1$ as required.

This completes the proof. □

We note that completely 0-simple semigroups satisfy the conditions of the above lemma. We are now ready to prove the main theorem of this section, the essence of which is that rational S-automata where S is a completely simple or completely 0-simple semigroup are no more powerful than G-automata where G is the maximal subgroup of S.

Theorem 6.2.3. *Let S be a completely simple or completely 0-simple semigroup with maximal non-zero subgroup G. Then*

$$F_{Rat}(S) = F_{Rat}(G) = F_1(G).$$

Proof. That $F_{Rat}(G) = F_1(G)$ is Theorem 4.2.3, while the inclusion $F_{Rat}(G) \subseteq F_{Rat}(S)$ is immediate. Hence, we need only prove that $F_{Rat}(S) \subseteq F_{Rat}(G)$. It follows easily from the Rees theorem that every completely simple semigroup S embeds in a completely 0-simple semigroup S' with the same maximal non-zero subgroup, so that $F_{Rat}(S) \subseteq F_{Rat}(S')$. Hence, it suffices to prove the result in the case that S is completely 0-simple.

Suppose, then, that S is completely 0-simple. By the Rees theorem, we may assume that S is a regular Rees matrix semigroup $M^0(G^0; I, J; P)$ where G is a

group. Suppose now that a language $L \subseteq \Sigma^*$ lies in $F_{Rat}(S)$. Let A be a rational S-automaton accepting L, with initial rational set $X_0 \subseteq S$ and terminal rational set $X_1 \subseteq S$. By Lemma 6.2.2, we may assume that $0 \notin X_0$ and $0 \notin X_1$.

Let C and D be automata over S accepting X_0 and X_1 respectively. Since C, D and A have only finitely many edges between them, we may choose finite subsets $I' \subseteq I$ and $J' \subseteq J$ such that the edge labels of C and D all lie in $I' \times G \times J'$, and the edge labels of A all lie in $(I' \times G \times J') \times \Sigma^*$.

For each $i \in I'$ and $j \in J'$, we let $X_{ij} = \{g \in G \mid (i, g, j) \in X_0\}$. By Proposition 6.1.1, each X_{ij} is a rational subset of G. It follows that

$$X'_{ij} \;=\; X_{ij} \times \{\epsilon\}$$

is a rational subset of $G \times \Sigma^*$; let C_{ij} be an automaton accepting X'_{ij}.

Similarly, for each $i \in I'$ and $j \in J'$ we define $Y_{ij} = \{g^{-1} \in G \mid (i, g, j) \in X_1\}$. By Propositions 6.1.1 and 2.2.6, Y_{ij} is a rational subset of G, and so

$$Y'_{ij} \;=\; Y_{ij} \times \{\epsilon\}$$

is a rational subset of $G \times \Sigma^*$; let D_{ij} be an automaton accepting Y'_{ij}.

Assume without loss of generality that the automaton A and all the automata C_{ij} and D_{ij} have disjoint state sets. We construct from these automata a G-automaton B with

- state set the union of the state sets of C_{ij} and D_{ij} (for $i \in I'$ and $j \in J'$) together with $I' \times Q \times J'$ where Q is the state set of A, and a new state q'_0;
- initial state q'_0;
- terminal states the terminal states of the automata D_{ij};
- all the edges of the automata C_{ij} and D_{ij};
- for each $i \in I'$ and $j \in J'$, an edge from q'_0 to the initial state of C_{ij} labelled $(1, \epsilon)$;

- for each $i \in I'$ and $j \in J'$, an edge from each terminal state of C_{ij} to (i, q_0, j) labelled $(1, \epsilon)$, where q_0 is the initial state for A;
- for each edge in A from p to q labelled $((i, g, j), w)$ and each $i' \in I'$ and $j' \in J'$, an edge from (i', p, j') to (i', q, j) labelled $(P_{j'i}g, w)$;
- for each $i \in I'$, $j \in J'$ and terminal state p of A, an edge from (i, p, j) to the initial state of D_{ij} labelled $(1, \epsilon)$.

Since I', J' and all the automata A, C_{ij} and D_{ij} are finite, it follows that the G-automaton B is finite. We now show that B accepts the language L.

Let $w \in L$. Then there exists a path through the automaton A labelled $((i, g, j), w)$ connecting the initial state with some terminal state (p_t say), such that

$$(i_0, g_0, j_0)(i, g, j) = (i', g', j') \in X_1$$

for some $(i_0, g_0, j_0) \in X_0$. Suppose this path has the form

$$\mathbf{q_0} \xrightarrow{((i_1,g_1,j_1),x_1)} \mathbf{q_1} \xrightarrow{((i_2,g_2,j_2),x_2)} \mathbf{q_2} \xrightarrow{((i_3,g_3,j_3),x_3)} \cdots \mathbf{q_{m-1}} \xrightarrow{((i_m,g_m,j_m),x_m)} \mathbf{q_m}$$

where q_0 is the initial state and $q_m = p_t$ is a terminal state of A and $w = x_1 \dots x_m$. Note that we must have $i' = i_0$, $j' = j_m$ and

$$g = g_1 P_{j_1 i_2} g_2 \dots P_{j_{m-1} i_m} g_m.$$

Now by construction, B has a path π_2 of the form

$$(\mathbf{i_0}, \mathbf{q_0}, \mathbf{j_0}) \xrightarrow{(P_{j_0 i_1} g_1, x_1)} (\mathbf{i_0}, \mathbf{q_1}, \mathbf{j_1}) \xrightarrow{(P_{j_1 i_2} g_2, x_2)} (\mathbf{i_0}, \mathbf{q_2}, \mathbf{j_2}) \xrightarrow{(P_{j_2 i_3} g_3, x_3)} \cdots$$
$$\cdots \xrightarrow{(P_{j_{m-1} i_m} g_m, x_m)} (\mathbf{i_0}, \mathbf{q_m}, \mathbf{j_m})$$

Moreover, from the fact that $(i_0, g_0, j_0) \in X_0$ we see that $g_0 \in X_{i_0 j_0}$, so that $(g_0, \epsilon) \in X'_{i_0 j_0}$. Hence, (g_0, ϵ) labels a path in $C_{i_0 j_0}$ from the initial state to a terminal state. Since the first part of the automaton B contains an exact copy of $C_{i_0 j_0}$, whose terminal states are connected in B to a copy of the initial state of A, it follows that (g_0, ϵ) labels a path π_1 in B from the initial state q'_0 to (i_0, q_0, j_0) where q_0 was the initial

state of A. Similarly, since $(i', g', j') \in X_1$ we deduce that $((g')^{-1}, \epsilon) \in Y_{i'j'} = Y_{i_0 j_m}$ so that B has a path π_3 from (i_0, q_m, j_m) to a terminal state labelled $((g')^{-1}, \epsilon)$.

Composing the paths π_1, π_2 and π_3, we see that B has a path from the initial state to a terminal state with label

$$(g_0 P_{j_0 i_1} g_1 P_{j_1 i_2} g_2 \dots P_{j_{m-1} i_m} g_m (g')^{-1}, x_1 x_2 \dots x_m)$$

But we know that $(i_0, g_0, j_0)(i, g, j) = (i', g', j')$, so we must have

$$g_0 P_{j_0 i_1} g_1 P_{j_1 i_2} g_2 \dots P_{j_{m-1} i_m} g_m = g'$$

and hence

$$g_0 P_{j_0 i_1} g_1 P_{j_1 i_2} g_2 \dots P_{j_{m-1} i_m} g_m (g')^{-1} = 1.$$

It follows that w is accepted by the G-automaton B, as required.

Conversely, suppose w is accepted by the G-automaton B. Then there is a path in B from the initial state to a terminal state labelled $(1, w)$. We deduce easily from the construction of B that this path must have the form $\pi_1 \pi_2 \pi_3$ where

- π_1 runs from the start state to some state (i_0, q_0, j_0) with label of the form (g_0, ϵ) for some $g_0 \in X_{i_0 j_0}$, so that $(i_0, g_0, j_0) \in X_0$;
- π_2 runs from (i_0, q_0, j_0) to a state (i_0, q_m, j_m) where q_m is a terminal state of A and
- π_3 runs from (i_0, q_m, j_m) to a terminal state with label $((g')^{-1}, \epsilon)$ where $(g')^{-1} \in Y_{i_0 j_m}$, so that $(i_0, g', j_m) \in X_1$.

Moreover, π_2 must have the form

$$(\mathbf{i_0}, \mathbf{q_0}, \mathbf{j_0}) \xrightarrow{(P_{j_0 i_1} g_1, x_1)} (\mathbf{i_0}, \mathbf{q_1}, \mathbf{j_1}) \xrightarrow{(P_{j_1 i_2} g_2, x_2)} (\mathbf{i_0}, \mathbf{q_2}, \mathbf{j_2}) \xrightarrow{(P_{j_2 i_3} g_3, x_3)} \dots$$
$$\dots \xrightarrow{(P_{j_{m-1}, i_m} g_m, x_m)} (\mathbf{i_0}, \mathbf{q_m}, \mathbf{j_m})$$

where, since the label of the entire path π is $(1, w)$, we must have $w = x_1 \dots x_m$ and $g_0 P_{j_0 i_1} g_1 \dots P_{j_{m-1} i_m} g_m (g')^{-1} = 1$, that is,

$$g_0 P_{j_0 i_1} g_1 \dots P_{j_{m-1} i_m} g_m = g'.$$

We deduce from the path above and the construction of B that A has a path

$$\mathbf{q_0} \xrightarrow{((i_1,g_1,j_1),x_1)} \mathbf{q_1} \xrightarrow{((i_2,g_2,j_2),x_2)} \mathbf{q_2} \xrightarrow{((i_3,g_3,j_3),x_3)} \cdots \mathbf{q_{m-1}} \xrightarrow{((i_m,g_m,j_m),x_m)} \mathbf{q_m}$$

Since q_0 and q_m are initial and terminal states of A respectively, it follows that A accepts (x, w) where

$$x = (i_1, g_1, j_1)(i_2, g_2, j_2) \dots (i_m, g_m, j_m).$$

But (i_0, g_0, j_0) lies in X_0 and

$$\begin{aligned}(i_0, g_0, j_0)x &= (i_0, g_0, j_0)(i_1, g_1, j_1) \dots (i_m, g_m, j_m) \\ &= (i_0, g_0 P_{j_0 i_1} g_1 \dots P_{j_{m-1} i_m} g_m, j_m) \\ &= (i_0, g', j_m)\end{aligned}$$

lies in X_1, from which we deduce that the rational S-automaton A accepts the word w, and so $w \in L$ as required. □

Bibliography

[1] V. Amar and G. Putzolu. Generalizations of regular events. *Inform. Confr.*, 8:56–63, 1965.

[2] H. Ayik and N. Ruskuc. Generators and relations of Rees matrix semigroups. *Proceedings of the Edinburgh Mathematical Society*, 42:481 –495, 1999.

[3] M. Benois. Parties rationnelles du groupe libre. *C. R. Acad. Sci. Paris*, Ser. A 269:1188–1190, 1969.

[4] J. Berstel. Memento sur les transductions rationnelles. In *Actes de l'Ecole de Printemps sur les langages algébriques*. Bonascre, Arige, 1973.

[5] J. Berstel. *Transductions and Context-Free Languages*. Teubner Studienbucher, Stuttgart, 1979.

[6] L. Boasson. Two iteration theorems for some families of languages. *Journal of Computer and System Sciences*, 7:583–596, 1973.

[7] R.V. Book, M. Jantzen, and C. Wrathall. Monadic Thue systems. *Theoretical Computer Science*, 19:231–251, 1982.

[8] R.V. Book and F. Otto. *String rewriting systems*. Springer Verlag, New York, 1993.

[9] N. Chomsky and M. P. Schutzenberger. The algebraic theory of context-free languages. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Languages*, pages 118–161. 1963.

[10] A.H. Clifford and G.B. Preston. *The Algebraic Theory of Semigroups*, volume 1. American Mathematical Society, 1961.

[11] J. M. Corson. Extended finite automata and word problems. *International Journal of Algebra and Computation*, 15(3):455–466, 2005.

[12] J. Dassow. *Grammars and regulated rewriting*, pages 249 – 275. Springer, 2004.

[13] J. Dassow and V. Mitrana. Finite automata over the free generated groups. *International Journal of Algebra and Computation*, 10(6):725–738, 2000.

[14] M. J. Dunwoody. The accessibility of finitely presented groups. *Inventiones Mathematicae*, 81(3):449–457, 1985.

[15] M. Elder. G-automata, counter languages and the Chomsky heirarchy. In *Groups St. Andrews 2005*, volume 339 of *London Mathematical Society Lecture Notes Series*, 2005.

[16] M. Elder, M. Kambites, and G. Ostheimer. On groups and counter automata. *International Journal of Algebra and Computation*, 18:1345 – 1364, 2006.

[17] M. Elder and A. Mintz. (private communication).

[18] G.Z. Elston and G. Ostheimer. On groups whose word problem is solved by a counter automaton. *Theoretical Computer Science*, 320(2-3), 2004.

[19] H. Fernau and R. Stiebe. Valence grammars with target sets. In S. Yu M. Ito, Gh. Paun, editor, *Words, Semigroups and Transductions*, pages 129–140. World Scientific, Singapore, 2001.

[20] H. Fernau and R. Stiebe. Sequential grammars and automata with valences. *Theoretical Computer Science*, 276:377–405, 2002.

[21] S. M. Gersten, D. Holt, and T. Riley. Isoperimetric inequalities for nilpotent groups. *Geometric and functional analysis*, 13(4):795–814, 2003.

[22] R.H. Gilman. Formal languages and infinite groups. In *Geometric and Computational Perspectives on Infinite Groups (Minneapolis, MN and New Brunswick, NJ, 1994), DIMACS Series*, volume 25 of *Discrete Mathematics and Theoretical Computer Science*, Providence RI, 1996. American Mathematical Society.

[23] R.H. Gilman and M. Shapiro. On groups whose word problem is solved by a nested stack automaton. `arXiv:math.GR/9812028`, 1998.

[24] E.S. Golod. Some problems of Burnside type. In *Proc. Internat. Congr. Math. (Moscow, 1966)*, pages 284–289, Moscow, 1968.

[25] E.S. Golod and I.R. Shafarevich. On the class field tower. *Izv. Akad. Nauk SSSR*, 28:261–272, 1964.

[26] S.A. Greibach. Remarks on blind and partially blind one-way multicounter machines. *Theoretical Computer Science*, 7(3):311–324, 1978.

[27] R.I. Grigorchuk. On Burnside's problem on periodic groups. *Functional Anal. Appl.*, 14(1):41–43, 1980.

[28] M. Gromov. Groups of polynomial growth and expanding maps. *Publications Mathematiques I.H.E.S.*, 53, 1981.

[29] Z. Grunschlag. *Algorithms in Geometric Group Theory.* PhD thesis, University of California at Berkeley, 1990.

[30] P. Hall. *The Edmonton Notes on Nilpotent Groups.* Queen Mary College Mathematics Notes, 1969.

[31] T. Herbst. On a subclass of context-free groups. *RAIRO Inform. Theor. Appl.*, 25(3):255–272, 1991.

[32] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[33] J.M. Howie. *Fundamentals of Semigroup Theory.* Clarendon Press, 1995.

[34] M. Kambites. Word problems recognisable by deterministic blind monoid automata. *Theoretical Computer Science*, 362(1), 2006.

[35] M. Kambites. Formal languages and groups as memory. *Communications in Algebra*, 37:193–208, 2009.

[36] M. Kambites. Small overlap monoids II: automatic structures and normal forms. *Journal of Algebra*, 321:2302–2316, 2009.

[37] M. Kambites, P.V. Silva, and B. Steinberg. On the rational subset problem for groups. *J. Algebra*, 309(2):622–639, 2007.

[38] R.C. Lyndon and P.E. Schupp. *Combinatorial Group Theory*. Springer, 2001.

[39] A. Mateescu and A. Salomaa. *Aspects of Classical Language Theory*, pages 175–252. Springer, 1997.

[40] V. Mitrana and R. Stiebe. The accepting power of finite automata over groups. In *New Trends in Formal Languages*, volume 1218 of *Lecture Notes in Computer Science*. Springer, Berlin, 1997.

[41] V. Mitrana and R. Stiebe. Extended finite automata over groups. *Discrete Applied Mathematics*, 108:287–300, 2001.

[42] D. E. Muller and P.E. Schupp. The theory of ends, pushdown automata, and second order logic. *Theoretical Computer Science*, 37:5175, 1985.

[43] D.E. Muller and P.E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. System Sci.*, 26(3):295–310, 1983.

[44] A. Yu. Ol'shanskii. The Novikov-Adyan theorem. *Mat. Sb. (N.S.)*, 118(2):203–235, 1982.

[45] Gh. Paun. A new generative device: valence grammars. *Rev. Roumaine Math. Pures Appl.*, XXV(6):911–924, 1980.

[46] M. Pelletier and J. Sakarovitch. Easy multiplications II. extensions of rational semigroups. *Information and Computation*, 88(1):18–59, 1990.

[47] V. Red'ko and L. Lisovik. Regular events in semigroups. *Problems of Cybernetics*, 37:155–184, 1980.

[48] D. Rees. On semi-groups. *Proceedings of the Cambridge Philosophical Society*, 36:387–400, 1940.

[49] E. Render and M. Kambites. Polycyclic and bicyclic valence automata. In *Language and Automata Theory and Applications 2008*, volume 5196 of *Lecture Notes in Computer Science*, pages 464 – 475. 2008.

[50] E. Render and M. Kambites. Rational subsets of polycyclic monoids and valence automata. *Information and Computation*, 207(11):1329 – 1339, 2009.

[51] E. Render and M. Kambites. Semigroup automata with rational initial and terminal sets. *Theoretical Computer Science*, 411(7-9):1004 – 1012, 2010.

[52] J. Sakarovitch. Easy multiplications. I. The realm of Kleene's theorem. *Information and Computation*, 74(3):173–197, 1987.

[53] A.K. Salomaa. Probabilistic and weighted grammars. *Information and Control*, 15:529–544, 1969.

[54] J.B. Stephen. Inverse monoids and rational subsets of related groups. *Semigroup Forum*, 46(1):98–108, 1993.

Index