COUNTING SUBWORDS AND OTHER RESULTS RELATED TO THE GENERALISED STAR-HEIGHT PROBLEM FOR REGULAR LANGUAGES

Thomas Bourne

Counting Subwords and Other Results Related to the Generalised Star-Height Problem for Regular Languages

Thomas Bourne

University of St Andrews

FOUNDED 1413

This thesis is submitted in partial fulfilment for the degree of PhD at the University of St Andrews

August 10th 2017

Declarations

Candidate's declarations

I, Thomas Bourne, hereby certify that this thesis, which is approximately 32000 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in September 2013 and as a candidate for the degree of PhD Mathematics in September 2013, the higher study for which this is a record was carried out in the University of St Andrews between 2013 and 2017.

Date: Signature of candidate:

Supervisors' declarations

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD Mathematics in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date: Signature of supervisor:

Date: Signature of supervisor:

Permission for publication

Abstract

The Generalised Star-Height Problem is an open question in the field of formal language theory that concerns a measure of complexity on the class of regular languages; specifically, it asks whether or not there exists an algorithm to determine the generalised star-height of a given regular language. Rather surprisingly, it is not yet known whether there exists a regular language of generalised star-height greater than one.

Motivated by a theorem of Thérien, we first take a combinatorial approach to the problem and consider the languages in which every word features a fixed contiguous subword an exact number of times. We show that these languages are all of generalised star-height zero. Similarly, we consider the languages in which every word features a fixed contiguous subword a prescribed number of times modulo a fixed number and show that these languages are all of generalised star-height at most one.

Using these combinatorial results, we initiate work on identifying the generalised star-height of the languages that are recognised by finite semigroups. To do this, we establish the generalised star-height of languages recognised by Rees zero-matrix semigroups over nilpotent groups of classes zero and one before considering Rees zero-matrix semigroups over monogenic semigroups.

Finally, we explore the generalised star-height of languages recognised by finite groups of a given order. We do this through the use of finite state automata and 'count arrows' to examine semidirect products of the form $A \rtimes \mathbb{Z}_r$, where A is an abelian group and $\mathbb{Z}_r$ is the cyclic group of order r.

Acknowledgements

Over the course of the past three and a half years, I have been fortunate enough to receive a phenomenal amount of support from my family, friends and colleagues. I would like to take this opportunity to say that I am eternally grateful to all of you for everything that you have done for me during my time as a research student. Special thanks are owed to the following people:

To my supervisors, Professor Nik Ruškuc and Dr Colva Roney-Dougal, for your guidance throughout the PhD process and for challenging me to push myself beyond my self-perceived mathematical limits. Without your support I doubt I would have made it to the end of this journey.

To my father Karl, my mother Chris and my sister Katie, whose unconditional love has transformed me into the person that I am today. I hope that I have done you proud and that I continue to do so from here on out.

To Rachael, for filling me with enthusiasm on the days where I felt deflated and for encouraging me to explain 'what it is that I actually do' in a way that you understand. Also, for being on the receiving end of many of my jokes and for listening to all of my (terrible) puns.

To my friends: Jenni, for being my gig companion and fellow gin enthusiast; Julius, for sharing my obsession with the gym and for allowing me to make fun of your native tongue; Alex, for pretending to like gin for fear of missing out on social interaction; Sascha, for agreeing to be my flatmate, for relating all of your Dad's antics, and for shower pictures from around the world; and Amy, for remaining my closest friend despite us living 460 miles apart.

To my office colleagues in The (Not Quite As) Ginger (As Before) Office and next door neighbours in The Office of Doom, for providing copious amounts of tea and cake alongside both thought-provoking discussion and mindless chatter.

To all of the swimmers at the University of St Andrews Swimming Club and all of the coaches and swimmers at Step Rock Amateur Swimming Club, for tolerating my grumpiness and for presenting me with so many wonderful opportunities to remain involved in the sport that I care so passionately about.

Contents

Chapter 1

Introduction and Preliminaries

The Generalised Star-Height Problem for regular languages is a long-standing problem in the field of formal language theory. Specifically, it asks whether or not there exists an algorithm to determine the generalised star-height of a given regular language. In particular, it is not yet known whether there exists a regular language with generalised star-height greater than or equal to one. In this thesis, we establish the generalised star-height of a wide range of regular languages by using new and existing combinatorial and algebraic techniques.

To ensure that this thesis is self-contained, we introduce the prerequisite mathematical knowledge required in Chapter 1. This includes all of the necessary notions from set theory, semigroup theory (including relevant information on monoids and groups), formal language theory and automata theory. We introduce both the (restricted) star-height problem and the generalised star-height problem before giving an outline of known results for both.

In Chapter 2, we take a combinatorial approach to the generalised star-height problem and, motivated by a theorem of Thérien, consider the contiguous subword ordering on the set of all words over a fixed alphabet. We define the regular languages $\mathrm{Count}(w,k)$ and $\mathrm{ModCount}(w,k,n)$ and split into two cases, first considering words over a unary alphabet before turning our attention to words over a non-unary alphabet. In each case, we show that the language $\mathrm{Count}(w,k)$ is of generalised star-height zero while the language $\mathrm{ModCount}(w,k,n)$ is of generalised star-height at most one. After this, we highlight the construction of the regular expressions found for $\mathrm{Count}(w,k)$ and $\mathrm{ModCount}(w,k,n)$ through a handful of examples before concluding the chapter with a discussion regarding the appropriate formulation of a variety of monoid languages for the ModCount family of languages.

In Chapter 3, we venture into the world of Rees zero-matrix semigroups and begin a classification of the languages that are recognised by finite semigroups. The ‘building blocks’ of finite semigroups are the finite 0-simple semigroups and the zero semigroups and, by The Rees Theorem (Theorem 3.1), a semigroup with zero is completely 0-simple if and only if it is isomorphic to a Rees zero-matrix semigroup over a group with regular sandwich matrix. As such, we investigate the languages recognised by Rees zero-matrix semigroups over the trivial group and languages recognised by Rees zero-matrix semigroups over finite abelian groups, concluding in each case that the generalised star-height of these languages is at most one. To get at languages recognised by Rees zero-matrix semigroups over finite abelian groups, we first consider languages recognised by Rees zero-matrix semigroups over finite cyclic groups. We then make use of properties of direct products, homomorphisms and projection maps and appeal to the Fundamental Theorem of Finite Abelian Groups. To complete this chapter, we take a slight detour and show that languages recognised by Rees zero-matrix semigroups over finite monogenic semigroups are also of generalised star-height at most one.

In Chapter 4, we examine which languages are recognised by finite groups of a given order. This work is motivated by a theorem of Pin, Straubing and Thérien (Theorem 4.1) which states that every language recognised by a finite group of order less than 12 is of generalised star-height at most one. Their proof relies on analysing semidirect products of the form $A \rtimes \mathbb{Z}_2$, where A is an abelian group and $\mathbb{Z}_2$ is the cyclic group of order 2. This is done through the use of ‘cyclic’ automata and counting arrows. We attempt to extend this work to semidirect products of the form $A \rtimes \mathbb{Z}_3$ by adhering to their proof strategy. Unfortunately, this ultimately fails due to an issue with unique factorisation of words into non-overlapping subwords. Nonetheless, insight into the problem is gained even though the result is not extended to languages recognised by finite groups of order 12 or higher.

Finally, in Chapter 5, we present some open questions that relate to both the work portrayed in this thesis and to that of others who have contributed to the problem at hand.

1.1 Set theory

Sets, subsets and examples

A *set* is a collection of objects; its members are referred to as *elements*. In general, we will denote sets by upper case letters and elements of sets by lower case letters. If x is an element of a set X then we write $x \in X$; otherwise, $x \notin X$. If a set has finitely many elements then it is possible to list them, and we do this using braces; for example,

$$X = \{1, 2, 3, 4, 5\}.$$

Often, we need to describe a set by stipulating a property P that its elements must satisfy. In this case, we write

$$\{x \mid x \text{ satisfies property } P\};$$

that is, the set of all x such that x satisfies property P. When a set has infinitely many members we can describe it using a property or, in circumstances where membership follows a clear pattern, through the use of ellipses; for example,

$$\{\dots, -4, -2, 0, 2, 4, \dots\}$$

denotes the set of even integers.

A set worthy of special mention is the *empty set*, which is denoted by $\emptyset$. It is the unique set containing no elements.

Throughout this thesis, we denote the set of integers $\{\dots, -1, 0, 1, \dots\}$ by $\mathbb{Z}$, the set of positive integers $\{1, 2, 3, \dots\}$ by $\mathbb{Z}^+$ and the set of non-negative integers $\{0, 1, 2, \dots\}$ by $\mathbb{N}$. We often refer to elements of $\mathbb{N}$ as *natural numbers*.

Two sets X and Y are *equal* if they contain precisely the same elements. A set Y is a *subset* of a set X if all the elements of Y are also elements of X and we write $Y \subseteq X$; otherwise, $Y \nsubseteq X$. If $Y \subseteq X$ and $X \neq Y$ then Y is a *proper* subset of X and we write $Y \subset X$. Importantly, $\emptyset \subseteq X$ and $X \subseteq X$ for every set X. It should also be noted that $\mathbb{Z}^+ \subset \mathbb{N} \subset \mathbb{Z}$.

In many cases, we prove that two sets X and Y are equal by showing that each set is a subset of the other; that is,

$$X = Y \Leftrightarrow X \subseteq Y \text{ and } Y \subseteq X.$$

Operations on sets

Given two sets X and Y, we can combine and compare them in the following ways. The *intersection* of X and Y is the set of all elements that belong to both

X and Y; that is,

$$X \cap Y = \{z \mid z \in X \text{ and } z \in Y\}.$$

The *union* of X and Y is the set of all elements that belong to X or Y; that is,

$$X \cup Y = \{z \mid z \in X \text{ or } z \in Y\}.$$

The word 'or' never implies exclusivity, so 'or' always means 'this set, that set or both'. The *relative complement* of Y with respect to X is the set of all elements that belong to X but not Y; that is,

$$X \setminus Y = \{z \mid z \in X \text{ and } z \notin Y\}.$$

These *boolean operations* satisfy a number of laws that we do not explicate here.

In this thesis, we will be interested in subsets of some fixed set U. We refer to U as the *universe* and define the set $X^c = U \setminus X$ to be the *(absolute) complement* of X. The following equalities hold for any sets X and Y:

$$(X \cup Y)^c = X^c \cap Y^c \qquad \text{and} \qquad (X \cap Y)^c = X^c \cup Y^c. \tag{1.1}$$

Collectively, these equalities are known as *de Morgan's Laws.*

Products of sets and binary relations

Given two sets X and Y, define the *Cartesian product* of X and Y by

$$X \times Y = \{(x, y) \mid x \in X \text{ and } y \in Y\};$$

that is, the set of all ordered pairs in which the first component is an element of X and the second component is an element of Y. This construction can be extended to any finite number of sets by repeated applications of the rule.

Given two sets X and Y, a *binary relation* ρ is a subset of the product $X \times Y$; that is, ρ is a collection of ordered pairs (x, y) in $X \times Y$. If (x, y) belongs to ρ then x and y are said to be *ρ-related.* We often use infix notation for relations and write $x \ \rho \ y$ to show that x and y are ρ-related.

When $X = Y$, relations can satisfy certain additional properties that can lead to further classification. A binary relation $\sim$ on X is:

- *reflexive* if $x \sim x$ for all x in X;
- *symmetric* if $x \sim y$ implies $y \sim x$ for all x and y in X;
- *antisymmetric* if $x \sim y$ and $y \sim x$ imply $x = y$ for all x and y in X; and,
- *transitive* if $x \sim y$ and $y \sim z$ imply $x \sim z$ for all x, y and z in X.

A *partial order* is a binary relation that is reflexive, antisymmetric and transitive, while an *equivalence relation* is a binary relation that is reflexive, symmetric and transitive. The *equivalence class* of x is defined by

$$[x] = \{y \in X \mid y \sim x\};$$

that is, the set of all y in X such that y is $\sim$-related to x. The equivalence classes *partition* the set X; that is, X can be written as a disjoint union of its equivalence classes.

Functions

Let X and Y be sets. A *function* φ from X to Y is a binary relation that satisfies the following rule: for every x in X there exists precisely one y in Y such that $x \; \varphi \; y$. We refer to X as the *domain* of φ, to Y as the *codomain* of φ and write $\varphi : X \to Y$. The set of all functions from X to Y is denoted by Y^X.

Let $\varphi : X \to Y$ be a function. An element x in X is an *argument* of the function and its corresponding y value in Y is the *image* of x under φ. We write functions on the right of their arguments; that is, $x\varphi = y$.

Let A be a subset of X. We define the *image* of A by

$$\mathrm{im}(A) = \{y \in Y \mid y = x\varphi \text{ for some } x \in A\};$$

that is, the set of all y in Y such that y is the image of some x in A. In the case where $A = X$, we refer to $\mathrm{im}(X)$ as the image of the function φ.

Now suppose that B is a subset of Y. We define the *preimage* of B by

$$B\varphi^{-1} = \{x \in X \mid x\varphi \in B\};$$

that is, the set of all x in X such that the image of x under φ is an element of B. It is important to note that

$$B\varphi^{-1} = \bigcup_{b \in B} b\varphi^{-1};$$

that is, the preimage of a set B is the union of the preimages of the individual elements of B.

A function $\varphi : X \to Y$ is *injective* if $a\varphi = b\varphi$ implies $a = b$ for all a and b in X; that is, distinct elements of the domain never get mapped to the same element of the codomain. The function φ is *surjective* if every element y in Y has a corresponding element x in X such that $x\varphi = y$; in other words, $\mathrm{im}(\varphi) = Y$. If a function is both injective and surjective then it is said to be *bijective.*

1.2 Semigroups, monoids and groups

Semigroups, monoids and groups

A *semigroup* is a set S equipped with a well-defined, associative binary operation $\circ : S \times S \to S$; that is, $x \circ (y \circ z) = (x \circ y) \circ z$ for all x, y and z in S. When the binary operation is clear, we denote the semigroup $(S, \circ)$ by S alone. We also drop the symbol denoting the binary operation and show it acting on pairs of elements by juxtaposition; that is, we write xy instead of $x \circ y$.

Let S be a semigroup. The cardinality of the underlying set, denoted by $|S|$, is the *order* of the semigroup. If x and y in S satisfy the equality $xy = yx$ then x and y *commute*. The semigroup S is *commutative* if every pair of elements commute.

If S has order at least two and there exists an element 0 in S satisfying the equalities $x0 = 0 = 0x$ for all x in S then 0 is a *zero element* of S and S is a *semigroup with zero*. It follows, by definition, that if a semigroup contains a zero element then it is unique. If S does not contain a zero element then we can adjoin a new element 0 to S and define $x0 = 0 = 0x$ for all x in $S \cup \{0\}$. The resulting semigroup is the *semigroup obtained from S by adjoining a zero*, which we denote by S^0.

In a similar vein, if there exists an element 1 in S satisfying the equalities $1x = x = x1$ for all x in S then 1 is an *identity element* of S and S is a *monoid* or *semigroup with identity*. We will usually denote a monoid by M. It follows, by definition, that if a semigroup contains an identity element then it is unique. If S does not contain an identity element then we can adjoin a new element 1 to S and define $1x = x = x1$ for all x in $S \cup \{1\}$. The resulting monoid is the *monoid obtained from S by adjoining an identity*, which we denote by S^1.

Let M be a monoid with identity 1. If for every m in M there exists n in M such that $mn = 1 = nm$ then M is a *group* and n is the *inverse* of m, which we denote by m^{-1}. The inverse of a group element is, necessarily, unique. We will usually denote a group by G. A commutative group is said to be *abelian.*

If A and B are subsets of a semigroup S then

$$AB = \{ab \mid a \in A \text{ and } b \in B\}.$$

Whenever $A = B$ we write A^2, noting that this refers to $\{a_1 a_2 \mid a_1, a_2 \in A\}$ and not $\{a^2 \mid a \in A\}$.

Substructures

A non-empty subset T of a semigroup S is a *subsemigroup* if it is closed with respect to the multiplication on S; that is, if xy lies in T for all x and y in T. If S is a monoid then a subsemigroup T of S is a *submonoid* if 1 is an element of T. Moreover, if a subsemigroup T of S forms a group with respect to the multiplication inherited from S then T is a *subgroup* of S. A semigroup in which every subgroup is trivial is said to be *aperiodic*.

A non-empty subset I of a semigroup S is called a *left ideal* if $SI \subseteq I$, a *right ideal* if $IS \subseteq I$ and a *(two-sided) ideal* if it is both a left ideal and a right ideal. By definition, all ideals of S are also subsemigroups but the converse does not hold. Ideals of S include S itself and, if S contains a zero element, $\{0\}$. An ideal I such that $\{0\} \subset I \subset S$ is said to be *proper*.

A semigroup without zero is said to be *simple* if it has no proper ideals. A semigroup S with zero is said to be 0*-simple* if $\{0\}$ and S are its only ideals and $S^2 \neq \{0\}$.

-morphisms and congruences

Let S and T be semigroups. A function $\varphi : S \to T$ is a *semigroup homomorphism* if $(st)\varphi = (s\varphi)(t\varphi)$ for all s and t in S. If S and T are monoids with identities 1_S and 1_T respectively, then φ is a *monoid homomorphism* if, in addition, $1_S\varphi = 1_T$. If, in addition, S and T are groups then φ is said to be a *group homomorphism* and, consequently, $(s^{-1})\varphi = (s\varphi)^{-1}$ for all s in S.

In all three cases, a *monomorphism* is an injective homomorphism, an *epimorphism* is a surjective homomorphism and an *isomorphism* is a bijective homomorphism. If there exists an isomorphism between S and T then they are *isomorphic*, and we write $S \cong T$.

A semigroup T *divides* a semigroup S if T is a homomorphic image of a subsemigroup of S; that is, if there exists a subsemigroup S' of S and an epimorphism $\varphi : S' \to T$.

Let S be a semigroup. A binary relation ρ on S is said to be *compatible* if $s_1 \ \rho \ t_1$ and $s_2 \ \rho \ t_2$ imply that $s_1s_2 \ \rho \ t_1t_2$ for all s_1, s_2, t_1 and t_2 in S. A compatible equivalence relation is a *congruence*.

If ρ is a congruence on a semigroup S then the *quotient semigroup* is the set $S/\rho = \{[s] \mid s \in S\}$ equipped with the multiplication defined by $[s][t] = [st]$.

Products of semigroups

Given two or more semigroups, it is possible to construct new semigroups in various different ways.

If S and T are semigroups then the *direct product* of S and T is the set $S \times T$ equipped with the binary operation

$$(s_1, t_1)(s_2, t_2) = (s_1 s_2, t_1 t_2),$$

where s_1 and s_2 are elements of S and t_1 and t_2 are elements of T. This construction can be extended to any finite number of semigroups by repeated applications of the rule.

Let S and T be semigroups. For clarity, we write S additively but do not assume that it is commutative. A *left action* of T on S is a function

$$T \times S \to S : (t, s) \mapsto ts$$

such that

$$t(s_1 + s_2) = ts_1 + ts_2$$
$$t_1(t_2 s) = (t_1 t_2)s$$

for all s, s_1 and s_2 in S and all t, t_1 and t_2 in T.

Given a left action of T on S, the *semidirect product* $S \rtimes T$ is the semigroup defined on $S \times T$ by the multiplication

$$(s_1, t_1)(s_2, t_2) = (s_1 + t_1 s_2, t_1 t_2),$$

where s_1 and s_2 are elements of S and t_1 and t_2 are elements of T.

For any two semigroups S and T, where S is written additively but not assumed to be commutative, the *wreath product* $S \wr T$ is the semigroup defined on $S^T \times T$ by the multiplication

$$(f_1, t_1)(f_2, t_2) = (f, t_1 t_2),$$

where f_1 and f_2 are elements of S^T, t_1 and t_2 are elements of T and f in S^T is defined by $tf = tf_1 + (tt_1)f_2$ for all t in T. Note that the wreath product $S \wr T$ is isomorphic to the semidirect product $S^T \rtimes T$.

Examples of groups

Let G be a group and let X be a subset of G. We say that X is a *generating set* for G if every element of G can be expressed as a combination of finitely many elements of X and their inverses; G is said to be *generated by* X and we write $G = \langle X \rangle$.

A *cyclic group* is a group that is generated by a single element. We denote the finite cyclic group of order n by $\mathbb{Z}_n$ and the infinite cyclic group by $\mathbb{Z}_\infty$. Every cyclic group is abelian and will often be written additively. The following theorem highlights the importance of cyclic groups.

Theorem 1.1 (Fundamental Theorem of Abelian Groups, [10, Theorem 2.2]). *Every finite abelian group is isomorphic to a direct product of finite cyclic groups of prime power order.*

Theorem 1.1 shows that every finite abelian group can be decomposed into a product of cyclic groups; that is, the cyclic groups are the 'building blocks' of the abelian groups.

The *dihedral group* of order $2n$, denoted by Dih_n, is the group of symmetries of a regular n-gon; for example, Dih_3 is the group of symmetries of an equilateral triangle. If ρ represents the rotation by $2\pi/n$ of the regular n-gon about its centre and σ represents any of the reflections of the n-gon then $\mathrm{Dih}_n = \langle \rho, \sigma \rangle$ subject to the conditions $\rho^n = 1$, $\sigma^2 = 1$ and $\rho\sigma = \sigma\rho^{-1}$.

For a group G, define the *lower central series* of G by

$$G_1 = G \qquad \text{and} \qquad G_{i+1} = [G_i, G],$$

where $[H, K]$ denotes the subgroup of G generated by all elements of the form $h^{-1}k^{-1}hk$, where h is in H and k is in K. The group G is said to be *nilpotent of class m* if $G_m \neq \{1\}$ and $G_{m+1} = \{1\}$.

It follows from the definition that the trivial group $\{1\}$ is the only nilpotent group of class 0 and that the non-trivial abelian groups are precisely the nilpotent groups of class 1. The smallest nilpotent group of class 2 is Dih_4, the dihedral group of order 8.

Pseudovarieties of monoids

A *pseudovariety of monoids* is a class of finite monoids that is closed under the taking of submonoids, homomorphic images and finite direct products. We denote pseudovarieties of monoids in bold font. Examples of pseudovarieties of monoids include: **Mon**, which consists of all finite monoids; **Grps**, which consists of all finite groups; and, **AbGrps**, which consists of all finite abelian groups. We define *pseudovarieties of semigroups* analogously.

For any collection of semigroups C, the pseudovariety of monoids $\mathbf{HSP}(C)$ *generated* by C is formed in the following way (see [13, Theorem 12.1.6] for more detail): first, we find all finite direct products of elements of C and form $\mathbf{P}(C)$; next, we find all submonoids of elements of $\mathbf{P}(C)$ and form $\mathbf{SP}(C)$; finally, we find all homomorphic images of elements of $\mathbf{SP}(C)$ and form $\mathbf{HSP}(C)$.

Given two pseudovarieties of monoids $\mathbf{V}$ and $\mathbf{W}$, we denote by $\mathbf{V} \rtimes \mathbf{W}$ the pseudovariety generated by all semidirect products of a monoid of $\mathbf{V}$ by a monoid of $\mathbf{W}$. Similarly, we denote by $\mathbf{V} \wr \mathbf{W}$ the pseudovariety generated by all wreath products of a monoid of $\mathbf{V}$ by a monoid of $\mathbf{W}$. Since $S \wr T$ is isomorphic to $S^T \rtimes T$ for any monoids S and T, we conclude the following:

Lemma 1.2 ([14, p. 709])**.** *The pseudovarieties* $\mathbf{V} \rtimes \mathbf{W}$ *and* $\mathbf{V} \wr \mathbf{W}$ *are equal.*

1.3 Formal languages

In this section, we aim to provide the reader with the necessary background information required from within the field of formal language theory. Many of the definitions and results that follow are standard. The interested reader should consult [8] and [13] for further details.

Alphabets, letters and words

An *alphabet* is a finite, non-empty set; its elements are *letters* (or *symbols*). From this point onwards, A will always denote an alphabet. The total number of letters in A is denoted by $|A|$. Two alphabets worthy of note are the unary alphabet $\{a\}$ and the binary alphabet $\{0, 1\}$, though we will often use $\{a, b\}$ for the latter so that, in order to aid understanding, all of our letters are letters in the everyday sense of the word.

A finite sequence of letters from A is a *word (over A).* Thus, formally, $(a_1, a_2, \ldots, a_r)$ is a word over some alphabet. For the sake of simplicity, we write words with their letters juxtaposed instead; that is, $a_1a_2 \ldots a_r$. It is possible to take a sequence of zero letters from A; the resulting word is the *empty word* and we denote it by ε. The set of all words over A is denoted by A^* and the set of all non-empty words over A is denoted by A^+. Note that A^* is the union of A^+ and the empty word.

The *length* of a word w, denoted by $|w|$, is the length of the sequence defining w; for example, $|ab| = 2$ and $|abbab| = 5$. The empty word is the unique word of length zero. If a is a letter from A then $|w|_a$ denotes the number of times a appears in w; for example, $|ab|_a = 1$ and $|abbab|_b = 3$. Note that

$$|w| = \sum_{a \in A} |w|_a.$$

Two words $v = a_1a_2 \dots a_r$ and $w = b_1b_2 \dots b_s$ over the same alphabet A are equal if and only if $r = s$ and $a_i = b_i$ for all $0 \leq i \leq r$; that is, if they are of the same length and feature exactly the same letters in exactly the same order.

Given two words v and w over the same alphabet A, we can form a new word by adjoining the letters of w to the end of v. The resulting word is the *concatenation of* v *and* w and is denoted by $v \cdot w$. When emphasis of the concatenation is not required, we drop the symbol denoting concatenation and write the resulting word as vw. It should be noted that, in general, concatenation of words is not commutative; that is, $vw \neq wv$. For example, $aab \cdot ba = aabba$ whereas $ba \cdot aab = baaab$. However, the order in which words are concatenated is unimportant when working over a unary alphabet. Note that the length of the resulting word vw after concatenation is equal to the sum of the lengths of the individual words; that is, $|vw| = |v| + |w|$.

Concatenation of words is an associative operation; that is, $u(vw) = (uv)w$ for all words u, v and w. The empty word plays an important role with regards to concatenation as $\varepsilon w = w = w\varepsilon$ for all words w. Because of these two facts, the set A^* of all words over A forms a monoid under concatenation; its identity element is the empty word. We refer to A^* as the *free monoid generated by* A. Similarly, A^+ is the *free semigroup generated by* A.

If w is a word then w^n, where $n \geq 1$, denotes the concatenation of w with itself n times. We define $w^0 = \varepsilon$. It follows that

$$w^m w^n = w^{m+n} \quad \text{and} \quad (w^m)^n = w^{mn}$$

hold for all natural numbers m and n.

Let u and v be elements of A^*. If $w = uv$ then u is a *prefix* of w and v is a *suffix* of w. The prefix u (respectively, suffix v) is *proper* if $u \neq w$ (respectively, $v \neq w$) and *non-empty* if $u \neq \varepsilon$ (respectively, $v \neq \varepsilon$). Any prefix of w that is also a suffix of w is a *border*. A border is *proper* if it is a proper prefix (and, therefore, also a proper suffix) and *non-empty* if it is not the empty word.

Languages and operations on languages

Given an alphabet A, a *monoid* (respectively, *semigroup*) *language* is a subset of A^* (respectively, A^+). For example, the set $K = \{a^n \mid n \in 2\mathbb{N}\}$ is a monoid language over the alphabet $\{a\}$ while the set $L = \{a^n b^n \mid n \in \mathbb{Z}^+\}$ is a semigroup language over the alphabet $\{a, b\}$.

If K and L are languages over the same alphabet then so too are $K \cup L$, $K \cap L$, $K \setminus L$ and $L^c = A^* \setminus L$; these operations are referred to as union, intersection, relative complement and complement, respectively. Union, intersection and complement are *boolean operations.*

We now define further operations that are specific to formal language theory. The *concatenation product* of K and L is the language consisting of all possible concatenations of a word from K with a word from L; that is,

$$KL = \{vw \mid v \in K \text{ and } w \in L\}.$$

Example 1.3. Examples of products of languages include:

1. For every language L, $L\emptyset = \emptyset = \emptyset L$.
2. For every language L, $\{\varepsilon\}L = L = L\{\varepsilon\}$.
3. If $K = \{a, ab\}$ and $L = \{b, ba, bb\}$ then

$$KL = \{ab, aba, abb, abba, abbb\}$$

and

$$LK = \{ba, bab, baa, baab, bba, bbab\}.$$

The third part of Example 1.3 shows that, in general, $KL \neq LK$. Moreover, KL and LK need not have the same cardinality.

The next language operation that we wish to define is that of (the *Kleene*) *star.* If L is a language then we define $L^0 = \{\varepsilon\}$, $L^{n+1} = L^nL$ and

$$L^* = \bigcup_{n \geq 0} L^i.$$

Example 1.4. Examples of Kleene stars of languages include:

1. $\emptyset^* = \{\varepsilon\}$.
2. $\{\varepsilon\}^* = \{\varepsilon\}$.
3. If $L = \{a^3\}$ then $L^* = \{\varepsilon, a^3, a^6, a^9, \dots\}$.
4. If $L = \{a, ab\}$ then $L^* = \{\varepsilon, a, ab, aa, aab, aba, abab, \dots\}$.

A related notion is that of the *(Kleene) plus* operation, which is defined by

$$L^+ = \bigcup_{n \geq 1} L^i.$$

Thus, $L^* = L^+ \cup \{\varepsilon\}$.

If K and L are two languages of A^* then the *left quotient* (or *residual*) of L by K is the language $K^{-1}L$ defined by

$$K^{-1}L = \{v \in A^* \mid \exists u \in K \text{ such that } uv \in L\}.$$

The *right quotient* (or *residual*) of L by K is defined analogously.

Example 1.5. Examples of left and right quotients of languages include:

1. For every language L, $\{\varepsilon\}^{-1}L = L = L\{\varepsilon\}^{-1}$.
2. For every language L containing ε, $L^{-1}\{\varepsilon\} = \{\varepsilon\} = \{\varepsilon\}L^{-1}$.
3. For every language L not containing ε, $L^{-1}\{\varepsilon\} = \emptyset = \{\varepsilon\}L^{-1}$.

By combining the above operations in different ways, we can write down a wide range of *expressions* that *represent* certain languages. In order to avoid the tedious nature of writing { and } repeatedly, we will omit them and use (and) as and when required for avoiding ambiguity. This is especially prevalent for singleton sets; the set $\{w\}$ will virtually always be written as w.

Example 1.6. Examples of languages over the alphabet $A = \{a, b\}$:

1. We can write the set of all words over A, namely A^*, as $(a \cup b)^*$. Here, brackets are included as $a \cup b^*$ represents an entirely different language. For comparison,

   $$A^* = (a \cup b)^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \dots\},$$

   while

   $$a \cup b^* = \{a, \varepsilon, b, bb, bbb, \dots\}.$$

   This example reflects the convention that the star operation takes precedence over the union operation.
2. The language K in which all words have prefix ab is represented by the expression $ab(a \cup b)^*$.
3. The language L in which all words have suffix ab is represented by the expression $(a \cup b)^*ab$.
4. The language in which all words have border ab, namely $K \cap L$, is represented by the expression

   $$ab(a \cup b)^* \cap (a \cup b)^*ab.$$

   Thus, an expression representing the intersection of two languages is given by the intersection of the two expressions representing each individual language. The same conclusion can be drawn for each of the other operations described above.

Hierarchy of languages

Languages form a containment hierarchy, as displayed below:

$$\begin{array}{c}\text{regular}\\ \cap\\ \text{context-free}\\ \cap\\ \text{context-sensitive}\\ \cap\\ \text{recursive}\\ \cap\\ \text{recursively enumerable}\end{array}$$

As can be seen, each class is a proper subclass of the class above it. For example, every regular language is also context-free but there exist context-free languages that are not regular. The archetypal example is that of the language $\{a^n b^n \mid n \geq 1\}$, which is context-free but not regular.

If we omit the class of recursive languages, then the hierarchy is attributed to the linguist Noam Chomsky. He first described the hierarchy in 1956 in terms of classes of formal grammars, a topic which we do not touch upon in this thesis.

It should be noted that the vast majority of languages do not fit into this hierarchy and, as such, the hierarchy cannot be considered as a complete classification of languages.

Regular languages via regular expressions

Within this thesis, we concern ourselves with the 'simplest' class of languages; that is, the regular languages. In older literature, this class of languages is also referred to as the class of *rational* languages. Depending on one's point of view, there are four different ways for defining the class of regular languages: grammars, as seen in the Chomsky hierarchy; a combinatorial approach; an algebraic approach; and, an approach via theoretical machines. In this thesis, we opt for the second of these approaches as our definition, though the third and fourth approaches will also be introduced and used throughout.

Given an alphabet A, we define the empty set, the empty word and each letter in A to be a *basic regular expression.* We recursively define new *regular expressions* in the following manner: if E and F are regular expressions then so are

(RE1) EF (product);
(RE2) $E \cup F$ (union); and,
(RE3) E^* (star).

Product, union and star are referred to as the *regular operators*, and they work in exactly the same way as in their definitions for languages. For example,

$$E^* = \bigcup_{n\geq 0} E^n.$$

The operations have an order of precedence: star, then product, then union. In Example 1.6, the first three examples all feature regular expressions whereas the fourth does not, as intersection is not a regular operator.

Every regular expression E represents a language, which we denote by $L(E)$. This leads to an important definition.

Definition 1.7. A language L is *regular* if there exists a regular expression E such that $L = L(E)$.

It is important to note that a regular language can be represented by more than one regular expression. For example, we have seen in Example 1.6 that the language A^* can be represented by the regular expression $(a \cup b)^*$. However, A^* can also be represented by the regular expression $(a^* \cup b^*)^*$.

Since we have also defined the intersection, relative complement and complement operations, it is natural to ask what happens to the class of regular languages under these operations. Interestingly, applying these operations to a regular language returns a language that is also regular. Specifically, the following is true.

Lemma 1.8 ([16, Corollary I.3.5])**.** *The class of regular languages is closed under complementation.*

Now, since every regular language is represented by a regular expression, it follows that we can use the complement operator in our expressions without introducing any new languages which are not regular. As such, any expression that uses the complement operator as well as product, union and star is referred to as a *generalised regular expression.* Note that, through de Morgan's Laws (Equation (1.1)), we are also free to use the intersection and relative complement operations within our generalised regular expressions.

Example 1.9. Examples of generalised regular expressions over the alphabet $A = \{a, b\}$:

1. The expression
$$(a \cup b)^*(ab \cup ba)^c$$
represents the language in which all words do not have suffix ab or ba; that is, all words that end with a double letter.

2. The expression
$$a(a \cup b)^* \setminus aa(a \cup b)^*$$
represents the language in which all words have prefix a but do not have prefix aa.
3. The expression
$$ab(a \cup b)^* \cap (a \cup b)^* ab$$
(as seen in Example 1.6) represents the language in which all words have border ab.

Recognition of a language by a monoid

By definition, a language is a subset of the free monoid generated by the set A. Because of this, it seems natural for there to be an algebraic approach to languages through the world of monoids. Indeed, the following definition can be made.

Definition 1.10. A language L of A^* is *recognised* by a monoid M if there exists a monoid homomorphism $\varphi : A^* \to M$ such that $L = (L\varphi)\varphi^{-1}$.

In literature, the following alternative definition may also be found.

Definition 1.11. A language L of A^* is *recognised* by a monoid M if there exists a monoid homomorphism $\varphi : A^* \to M$ and a subset P of M such that $L = P\varphi^{-1}$.

In either case, if there exists a monoid M such that L is recognised by M then L is said to be *recognisable.*

In the following lemma, we show the above two definitions are equivalent.

Lemma 1.12. *Let X and Y be sets and $\varphi : X \to Y$ be a function. For any subset L of X, it follows that $L = (L\varphi)\varphi^{-1}$ if and only if $L = P\varphi^{-1}$ for some subset P of Y.*

Proof. If $L = (L\varphi)\varphi^{-1}$ then $L = P\varphi^{-1}$ with $P = L\varphi$. Conversely, note that L is always a subset of $(L\varphi)\varphi^{-1}$, since if $l \in L$ then $l\varphi \in L\varphi$ and therefore $l \in \{x \in X \mid x\varphi \in L\varphi\} = (L\varphi)\varphi^{-1}$. Now, if $L = P\varphi^{-1}$ for some subset P of Y then

$$\begin{aligned} w \in (L\varphi)\varphi^{-1} &\Rightarrow w\varphi \in L\varphi \\ &\Rightarrow w\varphi = v\varphi \text{ for some } v \in L \\ &\Rightarrow w\varphi \in P \text{ (since } v \in L \text{ if and only if } v\varphi \in P) \end{aligned}$$

$$\Rightarrow w \in P\varphi^{-1} = L.$$

Hence $(L\varphi)\varphi^{-1} \subseteq L$ and, therefore, $L = (L\varphi)\varphi^{-1}$. □

In this thesis, we will allow ourselves the freedom to use the definition that is more convenient in each given situation.

The following characterisation of regular languages is, on rare occasions, used as its 'algebraic' definition.

Theorem 1.13 ([8, Theorem 3.1.4])**.** *A language is regular if and only if it is recognised by a finite monoid.*

A language may be recognised by more than one monoid, so it is natural to ask what the 'smallest' monoid recognising a given language is. In this setting, 'smallest' refers to the cardinality of the monoid recognising the language.

Syntactic monoids

Let L be a language over an alphabet A. The *syntactic congruence* of L, denoted by σ_L, is defined by $x \ \sigma_L \ y$ if and only if

$$uxv \in L \Leftrightarrow uyv \in L$$

for all u and v in A^*. The monoid $\mathrm{Syn}(L) = A^*/\sigma_L$ is the *syntactic monoid* of L.

The syntactic monoid of a language is the smallest monoid recognising said language. This is captured in the following lemma.

Lemma 1.14 ([8, Theorem 3.1.6])**.** *Let L be a language. A monoid M recognises L if and only if* $\mathrm{Syn}(L)$ *divides M.*

More generally, we can use the fact that division of monoids is a transitive relation (that is, if M divides N and N divides P then M divides P) to prove the following corollary.

Corollary 1.15. *Let M and N be monoids and let L be a language. If M recognises L and M divides N then N recognises L.*

Combining Theorem 1.13 and Lemma 1.14 results in the following theorem.

Theorem 1.16 ([8, Theorem 3.1.4])**.** *A language L is regular if and only if* $\mathrm{Syn}(L)$ *is finite.*

Varieties of languages

A *variety of monoid languages* L is a family of languages L_A, where A ranges over all alphabets, such that the following conditions hold:

(VL1) for each alphabet A, the set of languages L_A is a subset of $\mathcal{P}(A^*)$ that is closed under the Boolean operations;

(VL2) for each alphabet A, each language L in L_A and each letter a in A, both $a^{-1}L$ and La^{-1} belong to L_A; and,

(VL3) if $\varphi : A^* \to B^*$ is a monoid homomorphism and L belongs to L_B then $L\varphi^{-1}$ belongs to L_A.

We can also define a *variety of semigroup languages* in an analogous way by replacing all occurrences of $*$ with $+$. It is important to note that the variety of semigroup languages and the variety of monoid languages are fundamentally different. This is due to the fact that in a variety of monoid languages, condition (VL3) allows us to 'erase' letters by replacing them with the empty word. However, this is not possible when considering varieties of semigroup languages.

In the following theorem, we establish the link between pseudovarieties of monoids and varieties of monoid languages. The result remains true when 'monoid' is replaced by 'semigroup'.

Theorem 1.17 (Eilenberg's Variety Theorem, [5, Theorem VII.3.4s])**.** *There exists a one-to-one correspondence between the collection of all pseudovarieties of monoids and the collection of all varieties of monoid languages.*

The connection in Eilenberg's Variety Theorem is established in the following way. Given a pseudovariety of monoids $\mathbf{M}$, the corresponding variety of monoid languages consists of all those languages whose syntactic monoid is in $\mathbf{M}$. Conversely, given a variety of monoid languages L, the corresponding pseudovariety of monoids is generated by the syntactic monoids of those languages in L. As an example, if we consider the trivial pseudovariety of monoids $\mathbf{Triv}$ then the corresponding variety of monoid languages is

$$\mathsf{L}(\mathbf{Triv}) = \{\{\emptyset, A^*\} \mid A \text{ is an alphabet}\}.$$

1.4 Generalised star-height problem

Let A be an alphabet and let E and F be regular expressions for some languages over A. The *star-height* $h(E)$ of a regular expression is defined recursively as follows:

- $h(\emptyset) = h(\varepsilon) = h(a) = 0$, where a is a letter from A;
- $h(EF) = h(E \cup F) = \max\{h(E), h(F)\}$; and,
- $h(E^*) = h(E) + 1$.

For a regular language L, we define the *star-height* of L, which we denote by $h(L)$, to be

$$h(L) = \min\{h(E) \mid E \text{ is a regular expression for } L\};$$

that is, the minimum star-height of all regular expressions representing L.

Example 1.18. Let $A = \{a, b\}$ be an alphabet and consider the expressions $E = (a \cup b)^*$ and $F = (a^* \cup b^*)^*$. We see that $h(E) = 1$ and $h(F) = 2$. However, E and F both represent the language $L = A^*$. Since there does not exist an expression of star-height zero representing this language, we conclude that $h(L) = 1$.

The following questions were posed by Eggan in 1963.

Question 1.19 (Star-Height Problem)**.** *Can all regular languages be expressed using regular expressions of limited star-height? If not, does there exist an algorithm to determine the star-height of a given regular language?*

The first of these questions was answered in the negative when Eggan [4, Corollary 3.1] illustrated the existence of regular languages of star-height n for all natural numbers n. Expressions for languages of star-height 1, 2 and 3 are as follows:

$$\begin{aligned} E_1 &= a_1^* \\ E_2 &= (a_1^* a_2^* a_3)^* \\ E_3 &= ((a_1^* a_2^* a_3)^* (a_4^* a_5^* a_6)^* a_7)^*. \end{aligned}$$

These expressions can be defined recursively. Note that the expression E_n requires an alphabet of size at least $2^n - 1$. In his concluding remarks, Eggan questioned whether a language of star-height n could be found for all natural numbers n in the case where A is a binary alphabet. This was answered positively by Dejean and Schützenberger [3] in 1966. The expressions are defined recursively by

$$E_1 = (ab)^* \qquad \text{and} \qquad E_{n+1} = (\underbrace{a \dots a}_{2^n} \cdot E_n \cdot \underbrace{b \dots b}_{2^n} \cdot E_n)^*.$$

Thus, explicitly, the expressions are

$$E_1 = (ab)^*$$

$$\begin{aligned}
E_2 &= (aa(ab)^*bb(ab)^*)^* \\
E_3 &= (aaaa(aa(ab)^*bb(ab)^*)^*bbbb(aa(ab)^*bb(ab)^*)^*)^* \\
&\vdots
\end{aligned}$$

The second question posed in Question 1.19 was answered positively by Hashiguchi [6, Theorem 4.2] in 1983, but the algorithm he provided was computationally impossible from a practical point of view. A more efficient algorithm was devised by Kirsten [11] in 2005 but this algorithm is still deemed to be practically infeasible.

By Lemma 1.8, we know that the class of regular languages is closed under complementation. This allows us to use generalised regular expressions in order to represent regular languages. Thus, we can extend the definition of star-height of a regular expression to that of the *generalised star-height* of a generalised regular expression by defining $h(E^c) = h(E)$. It follows by de Morgan's Laws (Equation (1.1)) that

$$\begin{aligned}
h(E \cap F) &= h((E^c \cup F^c)^c) \\
&= h(E^c \cup F^c) \\
&= \max\{h(E^c), h(F^c)\} \\
&= \max\{h(E), h(F)\}
\end{aligned}$$

and

$$h(E \setminus F) = h(E \cap F^c) = \max\{h(E), h(F^c)\} = \max\{h(E), h(F)\}.$$

We define the *generalised star-height* of a regular language L as in the restricted case; that is,

$$h(L) = \min\{h(E) \mid E \text{ is a generalised regular expression for } L\}.$$

We can now pose the same questions as in Question 1.19:

Question 1.20 (Generalised Star-Height Problem)**.** *Does there exist an algorithm to determine the generalised star-height of a regular language? In particular, does there exist a language of generalised star-height greater than one?*

As implied by the phrasing of the questions above, the generalised star-height problem is considerably more difficult to solve than the star-height problem; remarkably, it is not yet known whether there exist languages with generalised star-height two or greater.

To end this section, we state some known results regarding the generalised star-height problem. This list is not exhaustive and further known results have been stated as and when they become relevant in future sections of this thesis.

Lemma 1.21. *For every natural number n, the set of languages of generalised star-height at most n is closed under set difference, concatenation product and the Boolean operations.*

Proof. This follows directly from the definition of generalised star-height. $\square$

Proposition 1.22 ([15, Proposition 4.1])**.** *For every natural number n, the set of languages of generalised star-height at most n is closed under left and right quotients.*

One of the first and by far the most influential result concerning generalised star-height is the following theorem.

Theorem 1.23 (Schützenberger, [17])**.** *A regular language is of generalised star-height zero if and only if its syntactic monoid is finite and aperiodic.*

Schützenberger's Theorem gives us an algorithm for determining whether or not a given language is of generalised star-height zero. This is most easily done through the use of finite state automata, which are introduced in Section 1.5.

Ideally, it would be most convenient if, given a generalised regular expression, we had a technique for removing stars without altering the language that it represents. In one of the most elementary cases, this can be done directly.

Lemma 1.24 ([15, Lemma 3.3])**.** *For any alphabet A and any subset B of A,*

$$A^* = \emptyset^c \qquad \text{and} \qquad B^* = A^* \setminus (A^*(A \setminus B)A^*) = (\emptyset^c(A \setminus B)\emptyset^c)^c.$$

Hence,

$$h(A^*) = h(B^*) = 0.$$

Proof. Let A be an alphabet and let B be a subset of A. The first equality is trivially true as the set of all words over A is equal to the complement of the set of no words over A.

Consider the second equality. We see that

$$\begin{aligned} w \in B^* &\Leftrightarrow w = \varepsilon \text{ or } w = b_1 b_2 \dots b_r \text{ for some } b_1, b_2, \dots, b_r \in B \\ &\Leftrightarrow w \text{ is not of the form } A^*aA^*, \text{ where } a \in A \setminus B \\ &\Leftrightarrow w \notin A^*(A \setminus B)A^* \\ &\Leftrightarrow w \in (A^*(A \setminus B)A^*)^c \\ &\Leftrightarrow w \in A^* \setminus (A^*(A \setminus B)A^*). \end{aligned}$$

Since neither the expression for A^* nor the expression for B^* contain any stars, we conclude that each has generalised star-height zero. $\square$

Though not a technique for removing stars, the Transfer Lemma ([15, Lemma 6.1]) can be used for reducing the height of a given generalised regular expression. Informally, it 'transfers' stars from one letter to another through the use of a substitution. We do not record this result here as it is not required in this thesis and only proves useful in a handful of cases.

1.5 Finite state automata

A *(finite state) automaton* is a quintuple $\mathcal{A} = (S, A, s_0, \delta, T)$, where

- S is a finite set of *states*;
- A is an *input alphabet*;
- s_0 in S is the *initial state*;
- $\delta : S \times A \to S$ is the *transition function*; and,
- $T \subseteq S$ is a set of *terminal states.*

In order to ensure that δ is a valid function, we insist that all of our automata are *complete* and *deterministic*, meaning that $\delta(s, a)$ is uniquely defined for all s in S and all a in A.

We represent automata using *transition diagrams*, which are special types of directed, labelled graphs. The vertices are labelled by elements of S and there exists an arrow from the vertex s to the vertex t labelled by the letter a whenever $\delta(s, a) = t$. The initial state is identified by an inward-pointing arrow and terminal states are identified by a double border.

Example 1.25. Define an automaton $\mathcal{A} = (S, A, s_0, \delta, T)$ by

- $S = \{0, 1\}$;
- $A = \{a, b\}$;
- $s_0 = 0$;
- $\delta(0, a) = 1$, $\delta(0, b) = 1$, $\delta(1, a) = 0$ and $\delta(1, b) = 0$; and,
- $T = \{0\}$.

The transition diagram of $\mathcal{A}$ is shown in Figure 1.1.

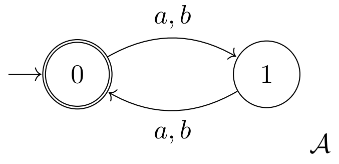

Figure 1.1: Transition diagram for the automaton $\mathcal{A}$ in Example 1.25.

For any given automaton, we can uniquely extend the transition function δ to the domain $S \times A^*$; that is, $\delta : S \times A^* \to S$. We define

(ETF1) $\delta(s, \varepsilon) = s$; and,
(ETF2) $\delta(s, wa) = \delta(\delta(s, w), a)$.

Thus, we can now deal with input words and not just input letters.

Let $\mathcal{A} = (S, A, s_0, \delta, T)$ be an automaton. A word w over A is *accepted* by $\mathcal{A}$ if there exists a path labelled by w beginning at the start state and ending at some terminal state; that is, if $\delta(s_0, w)$ lies in T. Otherwise, w is *rejected*. We define the language *recognised* by $\mathcal{A}$, which we denote by $L(\mathcal{A})$, to be

$$L(\mathcal{A}) = \{w \in A^* \mid \delta(s_0, w) \in T\};$$

that is, the set of all words w that are accepted by $\mathcal{A}$.

Example 1.26. The automaton in Example 1.25 accepts the words

$$\varepsilon, aa, ab, ba, bb, aaaa, aaab, aaba, aabb, \ldots$$

and rejects the words

$$a, b, aaa, aab, aba, abb, baa, bab, bba, bbb, aaaaa, \ldots.$$

Thus, $L(\mathcal{A}) = (aa \cup ab \cup ba \cup bb)^*$; that is, the set of all words of even length.

At this point, we make note of the following theorem of Kleene, which ties together the concept of regular languages to that of finite state automata.

Theorem 1.27 (Kleene, [12]). *A language is recognisable if and only if it is regular.*

Theorem 1.27 establishes a connection between the concept of regular languages and the concept of recognisable languages; indeed, it shows that they are equivalent notions. This means that we can construct automata for languages in order to prove that they are regular and can tackle the generalised star-height problem via automata too.

Let L be a recognisable language. An automaton $\mathcal{A}$ is said to be *minimal (for L)* if $\mathcal{A}$ recognises L and any other automaton recognising L has at least as many states as $\mathcal{A}$. It is clear from this definition that every recognisable language has an associated minimal automaton, and this automaton is unique up to isomorphism.

Let $\mathcal{A} = (S, A, s_0, \delta, T)$ be an automaton. For each w in A^*, define a function

$$\tau_w : S \to S : s \mapsto \delta(s, w).$$

The set of all such functions, which we denote by $T(\mathcal{A})$, is called the *transition monoid* of $\mathcal{A}$; that is,

$$T(\mathcal{A}) = \{\tau_w \mid w \in A^*\}.$$

The identity element of $T(\mathcal{A})$ is τ_ε.

Theorem 1.28 ([13, Theorem 9.4.3])**.** *Let L be a recognisable language. The transition monoid of the minimal automaton for L is isomorphic to the syntactic monoid of L.*

A related notion that we will make use of in Chapter 2 is that of a transducer. A *(finite state) transducer* is a sextuple $\mathcal{A} = (S, A, B, s_0, \delta, \gamma)$, where

- S is a finite set of *states*;
- A is an *input alphabet*;
- B is an *output alphabet*;
- s_0 in S is the *initial state*;
- $\delta : S \times A \to S$ is the *transition function*; and,
- $\gamma : S \times A \to B^*$ is the *output function.*

In order to ensure that δ is a valid function, we insist that all of our transducers are complete and deterministic.

As with automata, we represent transducers using transition diagrams where the label "$a|w$" on an arrow means "read the input letter a and output the word w".

Example 1.29. Define a transducer $\mathcal{A} = (S, A, B, s_0, \delta, \gamma)$ by

- $S = \{0, 1\}$;
- $A = \{a, b\}$;
- $B = \{a, b\}$;
- $s_0 = 0$;
- $\delta(0, a) = 1$, $\delta(0, b) = 1$, $\delta(1, a) = 1$ and $\delta(1, b) = 1$; and,
- $\gamma(0, a) = b$, $\gamma(0, b) = a$, $\gamma(1, a) = a$ and $\gamma(1, b) = b$.

The transition diagram of $\mathcal{A}$ is shown in Figure 1.2.

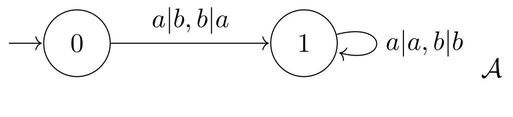

Figure 1.2: Transition diagram for the automaton $\mathcal{A}$ in Example 1.29.

For any non-empty word w over A, this transducer replaces the first letter of w with the other letter in A and leaves the rest of the word intact; for example, on input aab, the transducer outputs bab.

For any given transducer, we can uniquely extend the domain of δ to $S \times A^*$ in exactly the same way as we do with automata. Moreover, we can uniquely extend the domain of γ to $S \times A^*$ by defining

(EOF1) $\gamma(s, \varepsilon) = \varepsilon$; and,
(EOF2) $\gamma(s, wa) = \gamma(s, w)\gamma(\delta(s, w), a)$.

The function $\sigma : A^* \to B^*$ *realised* by $\mathcal{A}$ is defined by $w\sigma = \gamma(s_0, w)$ and a *sequential function* is a function realised by such a transducer.

Chapter 2

Counting Subwords

2.1 Motivation

Our motivation for counting subwords stems from a theorem of Thérien (see Theorem 2.1 below), first proved in 1983. Thérien's Theorem establishes a connection between counting scattered subwords and recognition of a language by a finite nilpotent group.

A word $w = a_1 a_2 \dots a_r$ is a *scattered subword* of a word v if there exist words $v_0, v_1, \dots, v_r$ over A such that $v = v_0 a_1 v_1 a_2 \dots a_r v_r$. We use the notation $\binom{v}{w}$ to denote the number of times w appears as a scattered subword of v. As an example, the words bc, ba and aa are all scattered subwords of $abaca$, with $\binom{abaca}{bc} = 1$, $\binom{abaca}{ba} = 2$ and $\binom{abaca}{aa} = 3$.

For every word w in A^+, every natural number k and every integer n greater than or equal to 2 with $0 \leq k < n$, we define the regular language $\mathrm{ScatModCount}(w, k, n)$ by

$$\mathrm{ScatModCount}(w, k, n) = \left\{ v \in A^* \mid \binom{v}{w} \equiv k \pmod{n} \right\};$$

that is, the set of words v over A such that w appears as a scattered subword of v precisely k modulo n times.

We can now state Thérien's Theorem:

Theorem 2.1 (Thérien, [18, Theorem 5])**.** *A language is recognised by a finite nilpotent group of class m if and only if it is a boolean combination of languages of the form* $\mathrm{ScatModCount}(w, k, n)$*, where $|w| \leq m$.*

Despite this characterisation, there are surprisingly few results concerning the generalised star-height of languages recognised by finite nilpotent groups.

This is due to the fact that determining the generalised star-height of the languages ScatModCount(w, k, n) is much harder than it first appears. The following results summarise what is known:

Theorem 2.2 (Henneman, [7]). *Every language recognised by a finite abelian group (that is, a finite nilpotent group of class one) is of generalised star-height at most one.*

Theorem 2.3 ([15, Theorem 7.3]). *Every language recognised by a finite nilpotent group of class two is of generalised star-height at most one.*

Very little is known about the generalised star-height of languages recognised by finite nilpotent groups of higher class. The following result showcases what can be established for a certain word form, and this result is used in the proof of the partial result for nilpotent groups of class three.

Proposition 2.4 ([15, Theorem 7.4]). *Let a and b be two distinct letters from an alphabet A. The generalised star-height of* ScatModCount(a^iba^j, k, n) *is at most one for all natural numbers i, j, k and n with $n \geq 2$ and $0 \leq k < n$.*

Theorem 2.5 ([15, Theorem 7.5]). *Let a, b and c be letters from an alphabet A. If $n \geq 2$ is a square-free integer then the generalised star-height of* ScatModCount(abc, k, n) *is at most one for all natural numbers k with $0 \leq k < n$.*

Occurrences of scattered subwords can be considered as a partial order on the set of words A^*. Indeed, we can define an order $\leq_{scat}$ by

$$v \leq_{scat} w \Leftrightarrow v \text{ is a scattered subword of } w.$$

A second partial order that we can place on A^* is that of occurrences of contiguous subwords. Define the order $\leq_{cont}$ by

$$v \leq_{cont} w \Leftrightarrow v \text{ is a contiguous subword of } w.$$

The concept of contiguous subword is defined in Section 2.2 and forms the basis of the remainder of this chapter.

The results found in Sections 2.3 and 2.4 have been published in [2].

2.2 Definitions

Let u, w and x be elements of A^*. If $v = uwx$ then w is a *contiguous subword* (often, *factor*) of v; for example, ε, a and bab are all contiguous subwords of *ababa*. From this point on, the word 'subword' will always means contiguous subword.

For every word w in A^+ and every word v in A^*, we denote the number of times that w appears as a subword of v by $|v|_w$. When w is a letter, say $w = a$, the notation $|v|_a$ coincides with its usual meaning; that is, the number of times the letter a appears in a word v.

For every word w in A^+ and every natural number k, we define the language $\text{Count}(w, k)$ by

$$\text{Count}(w, k) = \{v \in A^* \mid |v|_w = k\};$$

that is, the set of words v over A such that w appears as a subword of v precisely k times. As such, we regard $\text{Count}(w, 0)$ as the set of all words that do not feature w as a subword. From this characterisation, we note that

$$v \in \text{Count}(w, 0) \Leftrightarrow v \in A^* \setminus A^*wA^* \Leftrightarrow v \in (A^*wA^*)^c \Leftrightarrow v \in (\emptyset^c w \emptyset^c)^c, \quad (2.1)$$

where the final equivalence follows by Lemma 1.24. Thus, for a fixed word w, $\text{Count}(w, 0)$ can be represented by a star-free expression and is, therefore, of generalised star-height zero.

In order to simplify the proofs of some of the forthcoming results, we introduce the language $\text{CountBorder}(w, k)$ which is defined by

$$\text{CountBorder}(w, k) = wA^* \cap \text{Count}(w, k) \cap A^*w;$$

that is, the set of words v over A such that w is a prefix of v, w is a suffix of v and w appears as a subword of v precisely k times.

In a similar manner, for every word w in A^+, every natural number k and every integer n greater than or equal to 2 with $0 \leq k < n$, we define the language $\text{ModCount}(w, k, n)$ by

$$\text{ModCount}(w, k, n) = \{v \in A^* \mid |v|_w \equiv k \pmod{n}\};$$

that is, the set of words v over A such that w appears as a subword of v precisely k modulo n times. When finding expressions for $\text{ModCount}(w, k, n)$, our general strategy is to first count k occurrences of the subword w and then repeat in multiples of n.

Throughout this chapter we will treat $\text{ModCount}(w, 0, n)$ as a special case. With slight abuse of notation, we see that

$$\text{ModCount}(w, 0, n) = \text{Count}(w, 0) \cup \text{ModCount}(w, n, n).$$

This equality is true as, on the right-hand side, the $\text{Count}(w, 0)$ term covers all words that contain precisely zero occurrences of w while the $\text{ModCount}(w, n, n)$

term covers all words that contain occurrences of w in positive multiples of n. The reverse inclusion is immediate.

Notice that $\mathrm{ModCount}(w, k, n)$ can be defined in terms of $\mathrm{Count}(w, k)$; explicitly,

$$\mathrm{ModCount}(w, k, n) = \bigcup_{m \geq 0} \mathrm{Count}(w, k + mn).$$

At first glance, it may appear that the generalised star-height of the language $\mathrm{ModCount}(w, k, n)$ is equal to the maximum star-height of $\mathrm{Count}(w, k + mn)$, where m ranges over all natural numbers. However, this is not the case as the union on the right-hand side is infinite and is, therefore, not a generalised regular expression.

The regularity of both $\mathrm{Count}(w, k)$ and $\mathrm{ModCount}(w, k, n)$ can be established directly by constructing finite state automata accepting the languages and appealing to Kleene's Theorem. Alternatively, regularity can be established by constructing regular expressions for the languages, as is the case in the results presented throughout this chapter.

2.3 Over a unary alphabet

Let $A = \{a\}$ be a unary alphabet and consider the word $w = a^r$, where r is a positive integer. It is well known that a language L over A is regular if and only if L is of the form $X \cup Y(a^s)^*$, where X and Y are finite sets and s is a natural number; see, for example, [16, Proposition II.2.3]. Thus, every language over a unary alphabet is of generalised star-height at most one. However, we want to find expressions of minimal generalised star-height for $\mathrm{Count}(a^r, k)$ and $\mathrm{ModCount}(a^r, k, n)$.

We begin by finding an expression for $\mathrm{Count}(a^r, k)$. If we consider an arbitrary word a^s, where s is a positive integer, then, with the exception of the final $r - 1$ letters, each a appearing in a^s is the start of an occurrence of a^r. It immediately follows that

$$\mathrm{Count}(a^r, k) = a^{r+k-1} \tag{2.2}$$

for $k \geq 1$, and, as an alternative to that previously established in Equation (2.1),

$$\mathrm{Count}(a^r, 0) = \varepsilon \cup a \cup \cdots \cup a^{r-1}. \tag{2.3}$$

Next, we find an expression for $\mathrm{ModCount}(a^r, k, n)$ for non-zero values of k. In order to do this, we first count k occurrences of the subword a^r and then repeat in multiples of n. Recalling the expression for $\mathrm{Count}(a^r, k)$ in Equation (2.2) we obtain

$$\mathrm{ModCount}(a^r, k, n) = a^{r+k-1}(a^n)^*. \tag{2.4}$$

An expression for the remaining language, namely $\mathrm{ModCount}(a^r, 0, n)$, is obtained by using similar reasoning, while keeping in mind the special nature of $\mathrm{Count}(a^r, 0)$ as in Equation (2.3); it yields

$$\mathrm{ModCount}(a^r, 0, n) = \varepsilon \cup a \cup \cdots \cup a^{r-1} \cup a^{r+n-1}(a^n)^*.$$

It should be noted that the expression for $\mathrm{ModCount}(a^r, k, n)$ actually only depends on $r+k$ and n. Thus, different combinations of the parameters r, k and n may lead to the same regular expression and, hence, the same language. For example, $\mathrm{ModCount}(a^3, 1, 4)$, $\mathrm{ModCount}(a^2, 2, 4)$ and $\mathrm{ModCount}(a, 3, 4)$ are all represented by the expression $a^3(a^4)^*$.

A combination of the above constitutes a proof for the following lemma:

Lemma 2.6. *Let $A = \{a\}$ be a unary alphabet. For every positive integer r, the language $\mathrm{Count}(a^r, k)$ is of generalised star-height zero and the language $\mathrm{ModCount}(a^r, k, n)$ is of generalised star-height at most one.* □

2.4 Over a non-unary alphabet

We now deal with the more complicated case of counting subwords over a non-unary alphabet. Our plan of attack here is to start by exploring the languages in which we count subwords of length one, then explore the languages in which we count subwords of length two, and so on.

When counting subwords of length two, we notice that our arguments can be generalised to a wider class of languages, namely those in which we count subwords that have the empty word as their only border and those in which we count subwords which are powers of a letter.

Counting subwords of length three is the first situation where a spanner is thrown in the works, and it is after this stage that we generalise our methods in order to explore the languages in which we count subwords of any fixed length.

2.4.1 Counting subwords of length 1

Counting subwords of length one equates to counting how many times a certain letter appears in a word. As such, the languages $\mathrm{ModCount}(a, k, n)$ and $\mathrm{ScatModCount}(a, k, n)$ are equal for a fixed letter a from an alphabet A and fixed k and n. Hence, counting subwords of length one is covered by the previously stated theorem of Henneman:

Theorem 2.7 (Henneman, [7]). *Every language recognised by a finite abelian group is of generalised star-height at most one.*

Proof. Let L be a language recognised by a abelian group. By Theorem 2.1, L is a boolean combination of languages of the form $\text{ModCount}(a, k, n)$, where a is a letter. An expression for $\text{ModCount}(a, k, n)$ is given by

$$\text{ModCount}(a, k, n) = (B^*a)^k((B^*a)^n)^*B^*,$$

where $B = A \setminus \{a\}$. By Lemma 1.24, $h(B^*) = 0$. Hence, the generalised star-height of $\text{ModCount}(a, k, n)$ is at most one and, in turn, the generalised star-height of L is at most one. □

2.4.2 Counting subwords of length 2

Next, we consider the case where the subword w under consideration is of length two. In this situation, every word is either of the form aa or of the form ab, where a and b are distinct letters from an alphabet A; that is, every word of length two is either a power of a letter or has the empty word as its maximal border. As such, we consider these two cases in full generality, beginning with the latter.

Suppose that w has maximal border ε, meaning that w does not overlap itself. Once we have started to read w we can continue reading it until it ends without worrying that another occurrence of w may have already begun. From Equation (2.1), we know that $\text{Count}(w, 0)$ can be represented by the star-free expression $(\emptyset^c w \emptyset^c)^c$. From this we can obtain an expression representing $\text{Count}(w, k)$ which is star-free:

$$\text{Count}(w, k) = (\text{Count}(w, 0) \cdot w)^k \cdot \text{Count}(w, 0). \tag{2.5}$$

As can be seen from this expression, we begin with a word from $\text{Count}(w, 0)$, which may be empty, and then count the k occurrences of the subword w, with each pair of occurrences 'padded' by a (possibly empty) word from $\text{Count}(w, 0)$. We finish with a word from $\text{Count}(w, 0)$, which, again, may be empty.

We now turn our attention to counting occurrences of w modulo n. Following our general strategy, an expression for $\text{ModCount}(w, k, n)$ which is of generalised star-height one is given by

$$(\text{Count}(w, 0) \cdot w)^k \cdot ((\text{Count}(w, 0) \cdot w)^n)^* \cdot \text{Count}(w, 0). \tag{2.6}$$

As can be seen from this expression, we begin with a word from $\mathrm{Count}(w, 0)$ and then count the first k occurrences of the subword w, with each pair of occurrences 'padded' by a word from $\mathrm{Count}(w, 0)$. After this, we allow the same expression to repeat in non-negative multiples of n before ending with a final word from $\mathrm{Count}(w, 0)$.

A combination of the above constitutes a proof for the following lemma:

Lemma 2.8. *Let A be a non-unary alphabet and let the word w over A have ε as its only border. Every language $\mathrm{Count}(w, k)$ is of generalised star-height zero and every language $\mathrm{ModCount}(w, k, n)$ is of generalised star-height at most one.* □

We now assume that our alphabet A contains at least two letters and analyse the case where the subword w under consideration consists of a power of a letter, say a. Specifically, we are interested in finding generalised regular expressions for $\mathrm{Count}(a^r, k)$ and $\mathrm{ModCount}(a^r, k, n)$, where r is a positive integer.

From Equation (2.1), we know that $\mathrm{Count}(a^r, 0)$ can be represented by the star-free expression $(\emptyset^c(a^r)\emptyset^c)^c$. In the case where $k > 0$, we find an expression representing $\mathrm{Count}(a^r, k)$ by first considering only those words where a^r is a border; that is, we consider $\mathrm{CountBorder}(a^r, k)$. Let $B = A \setminus \{a\}$. We think of B as a set of 'buffers' that stop us from 'accidentally' reading two 'a's in a row. This is important as letters may appear as a component of more than one subword and the buffers are used to mark the points where we stop reading powers of a. We also define the subset W of A^* by

$$W = B \cup (B \cdot \mathrm{Count}(a^r, 0) \cdot B),$$

which is the set of non-empty words that do not feature a^r as a subword and neither start nor end with a. It is useful to think of elements of W as 'wedges', separating the strings that feature a^r from one another. Note that the individual components of W are all star-free expressions which implies that W is a language of generalised star-height zero.

A general formula for $\mathrm{CountBorder}(a^r, k)$ is given in the following lemma:

Lemma 2.9. *Let A be a non-unary alphabet and let a be a letter from A. For any positive integer r,*

$$\mathrm{CountBorder}(a^r, k) = \bigcup_{j=1}^{k} \bigcup_{\substack{k_1, k_2, \ldots, k_j \geq r \\ k_1 + k_2 + \cdots + k_j = k + (r-1)j}} a^{k_1} W a^{k_2} W \ldots W a^{k_j}.$$

Hence, $\mathrm{CountBorder}(a^r, k)$ *is of generalised star-height zero.*

Proof. Consider an arbitrary word v in $\mathrm{CountBorder}(a^r, k)$. Let $a^{k_1}, \dots, a^{k_j}$ be the maximal subwords of v that are powers of a and have length greater than or equal to r. Note that a^{k_1} must be a prefix of v as v starts with a^r, and, likewise, a^{k_j} must be a suffix. Hence, we have a decomposition $v = a^{k_1}v_1a^{k_2}v_2 \dots v_{j-1}a^{k_j}$, where, necessarily, $v_1, \dots, v_{j-1}$ belong to W. Indeed, if each v_i did not belong to the set W then it would either be the letter a or would have border a and contain a power of a of length greater than r, both of which would contradict the maximality of the a^{k_i} subwords. Furthermore, each a^{k_i} contains precisely $k_i - r + 1$ occurrences of a^r by Equation (2.2). Since all of the occurrences of a^r appear as subwords of a^{k_i}, we must have

$$k = |v|_{a^r} = \sum_{i=1}^{j} |a^{k_i}|_{a^r} = \sum_{i=1}^{j}(k_i - r + 1) = k_1 + \cdots + k_j - (r-1)j,$$

and so v belongs to the right-hand side.

Conversely, consider an arbitrary word v from the right-hand side. We can factorise v as

$$v = a^{k_1}v_1a^{k_2}v_2 \dots v_{j-1}a^{k_j},$$

where each v_i is an element of W and each k_i is greater than or equal to r with $k_1 + k_2 + \cdots + k_j = k + (r-1)j$. Since each k_i is greater than or equal to r, a^r is both a prefix and a suffix of v and

$$\begin{aligned}
|v|_{a^r} &= |a^{k_1}v_1a^{k_2}v_2 \dots v_{j-1}a^{k_j}|_{a^r} \\
&\geq |a^{k_1}|_{a^r} + |a^{k_2}|_{a^r} + \cdots + |a^{k_j}|_{a^r} \\
&= (k_1 - r + 1) + (k_2 - r + 1) + \cdots + (k_j - r + 1) \\
&= k_1 + k_2 + \cdots + k_j - (r-1)j \\
&= k.
\end{aligned}$$

Hence, v contains at least k occurrences of a^r as a subword. Moreover, by the definition of W, there can be no further occurrences of a^r as a subword. We conclude that v contains precisely k occurrences of a^r as a subword and that v is an element of $\mathrm{CountBorder}(a^r, k)$.

Finally, we note that the right-hand side is a generalised regular expression since both unions are finite. Since the expression is star-free, it follows that $\mathrm{CountBorder}(a^r, k)$ is of generalised star-height zero. $\square$

Now, we make use of the above result to prove that $\mathrm{Count}(a^r, k)$ is of generalised star-height zero.

Proposition 2.10. *Let A be a non-unary alphabet and let a be a letter from A. For any positive integer r,* $\mathrm{Count}(a^r, k)$ *is expressed by*

$$(\varepsilon \cup (\mathrm{Count}(a^r, 0) \cdot B)) \cdot \mathrm{CountBorder}(a^r, k) \cdot ((B \cdot \mathrm{Count}(a^r, 0)) \cup \varepsilon).$$

Hence, $\mathrm{Count}(a^r, k)$ *is of generalised star-height zero.*

Proof. First, note that all k occurrences of a^r appear in the $\mathrm{CountBorder}(a^r, k)$ term. In order to not introduce any further occurrences of k, this term can be preceded by either the empty word or a word that does not contain a^r as a subword; that is, a word from $\mathrm{Count}(a^r, 0)$. However, since words in $\mathrm{Count}(a^r, 0)$ have the potential to end with a power of a, we must utilise a 'buffer' from the set B. A dual argument deals with potential suffices. Since each of the components of the expression are star-free, $\mathrm{Count}(a^r, k)$ must be of generalised star-height zero. □

We now turn our attention to counting occurrences of a^r modulo n. Again, we make use of our general strategy by counting the first k occurrences of a^r using the expression found above for $\mathrm{CountBorder}(a^r, k)$ and then counting occurrences of a^r in multiples of n. We then add on prefixes and suffices as appropriate (as in Proposition 2.10) in order to establish an expression for $\mathrm{ModCount}(a^r, k, n)$.

Having used $\mathrm{CountBorder}(a^r, k)$ to count the first k occurrences of a^r, we note that the suffix a^{r-1} has the potential to be a component of a new occurrence of a^r if the part of the word immediately following a^{r-1} begins with an a. Similarly, the suffix a^{r-2} immediately followed by an a^2 leads to another occurrence of a^r, and so on. In order to take these possibilities into account, let $\mathrm{Multiple}(a^r, n)$ denote the language whose words contain precisely n occurrences of the subword a^r when left concatenated by a^{r-1} and also have suffix a^r:

$$\mathrm{Multiple}(a^r, n) = \{w \in A^* \mid |a^{r-1}w|_{a^r} = n \text{ and } w = w'a^r \text{ for some } w \in A^*\}.$$

The significance of the assumption about the suffix a^r is that every count stops precisely when the nth occurrence of a^r is met, and that this suffix 'feeds into' the next group of occurrences of a^r.

Lemma 2.11. *Let A be a non-unary alphabet and let a be a letter from A. For any positive integer r,*

$$\mathrm{Multiple}(a^r, n) = a^n \cup \bigcup_{i=0}^{n-1} a^i W \cdot \mathrm{CountBorder}(a^r, n-i).$$

Hence, $\mathrm{Multiple}(a^r, n)$ *is of generalised star-height zero.*

Proof. Consider an arbitrary word v in $\mathrm{Multiple}(a^r, n)$. If $v = a^s$ for some positive integer s then

$$n = |a^{r-1}v|_{a^r} = |a^{r-1}a^s|_{a^r} = |a^{r+s-1}|_{a^r} = s$$

by Equation (2.2), and hence $v = a^n$. Otherwise, we can decompose v as

$$v = a^{k_1}v_1a^{k_2}v_2 \ldots v_{j-1}a^{k_j},$$

where,

$$v_1, \ldots, v_{j-1} \in W, \qquad k_1 \geq 0 \qquad \text{and} \qquad k_2, \ldots, k_j \geq r.$$

The maximal subwords of $a^{r-1}v$ that are powers of a of exponent greater than or equal to r are $a^{r-1}a^{k_1} = a^{r+k_1-1}$ (provided that $k_1 > 0$) and $a^{k_2}, \ldots, a^{k_j}$. Furthermore, our decomposition of v can be used to split $a^{r-1}v$ as $a^{r-1}v = xy$, where $x = a^{r+k_1-1}v_1$ and $y = a^{k_2}v_2 \ldots v_{j-1}a^{k_j}$. Suppose that x contains i occurrences of a^r. Then

$$i = |a^{r+k_1-1}v_1|_{a^r} = |a^{r+k_1-1}|_{a^r} = k_1$$

by Equation (2.2). Moreover, y must contain the remaining $n-i$ occurrences of a^r and has a^r as a border. Hence, y belongs to $\mathrm{CountBorder}(a^r, n-i)$. Thus, v belongs to $a^iW \cdot \mathrm{CountBorder}(a^r, n-i)$ and therefore belongs to the union on the right-hand side.

Conversely, consider an arbitrary word v from the right-hand side. If $v = a^n$ then

$$|a^{r-1}v|_{a^r} = |a^{r-1}a^n|_{a^r} = |a^{r+n-1}|_{a^r} = n,$$

by Equation (2.2), and $a^{r-1}v$ has suffix a^r. Hence, v belongs to $\mathrm{Multiple}(a^r, n)$. Otherwise, v is of the form a^iv_0x, where

$$0 \leq i \leq n-1, \qquad v_0 \in W \qquad \text{and} \qquad x \in \mathrm{CountBorder}(a^r, n-i).$$

Thus,

$$\begin{aligned} |a^{r-1}v|_{a^r} &= |a^{r-1}a^iv_0x|_{a^r} \\ &\geq |a^{r+i-1}|_{a^r} + |x|_{a^r} \\ &= i + (n-i) \\ &= n. \end{aligned}$$

Hence, $a^{r-1}v$ contains at least n occurrences of a^r as a subword. Moreover, by the definition of W, there can be no further occurrences of a^r as a subword.

Also, v has suffix a^r since x belongs to $\mathrm{CountBorder}(a^r, n-i)$. We conclude that $a^{r-1}v$ contains precisely n occurrences of a^r as a subword and that v is an element of $\mathrm{Multiple}(a^r, n)$.

Finally, we note that the right-hand side is a regular expression since the union is finite. As the expression is star-free, it follows that $\mathrm{Multiple}(a^r, n)$ is of generalised star-height zero. $\square$

We now combine the results presented above in order to deal with the language $\mathrm{ModCount}(a^r, k, n)$. An expression representing $\mathrm{ModCount}(a^r, k, n)$, where k is a positive integer, is given by

$$(\varepsilon \cup (\mathrm{Count}(a^r, 0) \cdot B)) \cdot \mathrm{CountBorder}(a^r, k) \cdot \\ \mathrm{Multiple}(a^r, n)^* \cdot ((B \cdot \mathrm{Count}(a^r, 0)) \cup \varepsilon),$$

and an expression representing $\mathrm{ModCount}(a^r, 0, n)$ is given, with slight abuse of notation, by

$$\mathrm{Count}(a^r, 0) \cup \mathrm{ModCount}(a^r, n, n).$$

Both of these expressions are of generalised star-height one, so the language $\mathrm{ModCount}(a^r, k, n)$ is of generalised star-height at most one. Note that the appended prefixes and suffices are justified in the same manner as that found in the proof of Proposition 2.10.

A combination of the above constitutes a proof for the following proposition:

Proposition 2.12. *Let A be a non-unary alphabet and let a be a letter from A. For every positive integer r, the language* $\mathrm{Count}(a^r, k)$ *is of generalised star-height zero and the language* $\mathrm{ModCount}(a^r, k, n)$ *is of generalised star-height at most one.* $\square$

2.4.3 Counting subwords of length 3

When the subword under consideration is of length three we are presented with a new hurdle to overcome. Up until this point, every word that we have considered has had either the empty word as its only border or has been a power of a letter. With words of length three, we encounter the word aba, which is neither a power of a letter nor a word with the empty word as its only border.

The possible types for words of length three are

$$aaa, \qquad aab, \qquad aba, \qquad abb \qquad \text{and} \qquad abc,$$

where a, b and c are distinct letters from A. Counting occurrences of the subword aaa is covered by Lemmas 2.6 and 2.12, while the subwords aab, abb and abc are covered by Lemma 2.8.

With the final type, namely aba, we must be more careful as it has a as a border, meaning that the suffix a can act as a prefix a in a new occurrence of the subword aba. For example, the word $abababa$ contains three occurrences of the subword aba. Below, we resolve this case in a similar fashion to that of Proposition 2.12 but do not provide all of the details of the proof.

Define W to be the set of words that are not b, do not have prefix ba, do not have suffix ab, and do not contain aba as a subword; that is,

$$W = (b \cup baA^* \cup A^*ab \cup A^*abaA^*)^c = (b \cup ba\emptyset^c \cup \emptyset^c ab \cup \emptyset^c aba\emptyset^c)^c.$$

Then, a general formula for $\text{CountBorder}(aba, k)$, where k is a positive integer, is given by

$$\text{CountBorder}(aba, k) = \bigcup_{j=1}^{k} \bigcup_{\substack{k_1,k_2,\ldots,k_j \geq 1 \\ k_1+k_2+\cdots+k_j=k}} a(ba)^{k_1}Wa(ba)^{k_2}W\ldots Wa(ba)^{k_j},$$

which is star-free, and the language $\text{Count}(aba, k)$, expressed by

$$(\emptyset^c aba\emptyset^c \cup \emptyset^c ab)^c \cdot \text{CountBorder}(aba, k) \cdot (ba\emptyset^c \cup \emptyset^c aba\emptyset^c)^c,$$

is of generalised star-height zero.

To find an expression for $\text{ModCount}(aba, k, n)$ we introduce the language

$$\text{Multiple}(aba, n) = \{w \in A^* \mid |aw|_{aba} = n \text{ and } w \text{ has suffix } aba\}.$$

A star-free expression representing $\text{Multiple}(aba, n)$ is given by

$$(ba)^n \cup \bigcup_{i=1}^{n-1} (ba)^i W \cdot \text{CountBorder}(aba, n-i).$$

Putting all of this together, an expression representing $\text{ModCount}(aba, k, n)$, where k is greater than 0, is given by

$$(\emptyset^c aba\emptyset^c \cup \emptyset^c ab)^c \cdot \text{CountBorder}(aba, k) \cdot \text{Multiple}(aba, n)^* \cdot (ba\emptyset^c \cup \emptyset^c aba\emptyset^c)^c,$$

and an expression representing $\text{ModCount}(aba, 0, n)$ is given, with slight abuse of notation, by

$$\text{Count}(aba, 0) \cup \text{ModCount}(aba, n, n).$$

This establishes that the language $\text{ModCount}(aba, k, n)$ is of generalised star-height at most one.

Hence, we have proven the following result:

Proposition 2.13. *Let A be an alphabet. For any word w in A^+ with $|w| \leq 3$, the language* $\mathrm{Count}(w,k)$ *is of generalised star-height zero and the language* $\mathrm{ModCount}(w,k,n)$ *is of generalised star-height at most one.* □

In Section 2.5, we refocus our efforts towards finding a general result that covers counting subwords of any length over any alphabet. Before doing this, we note that Proposition 2.13 can also be proved using existing theoretical results.

Let A and $B = \{b\}$ be alphabets, and consider the languages

$$L = \mathrm{ModCount}(b,k,n) = b^k(b^n)^*$$

over B and $K = \mathrm{ModCount}(w,k,n)$ over A. Define a function

$$f_w : A^* \to B^* : v \mapsto b^{|v|_w}.$$

Consider the preimage of L under f_w. We see that,

$$\begin{aligned} Lf_w^{-1} &= \{v \in A^* \mid vf_w \in L\} \\ &= \{v \in A^* \mid b^{|v|_w} \in b^k(b^n)^*\} \\ &= \{v \in A^* \mid |v|_w = k + mn \text{ for some } m \in \mathbb{N}\} \\ &= K. \end{aligned}$$

Note that for all words w, the function f_w is sequential. For example, a transducer realising f_{aba} is shown in Figure 2.1. In order to improve readability, edges labelled with $c|\varepsilon$, where c lies in $A \setminus \{a,b\}$, have been removed; all such edges point directly to the initial state.

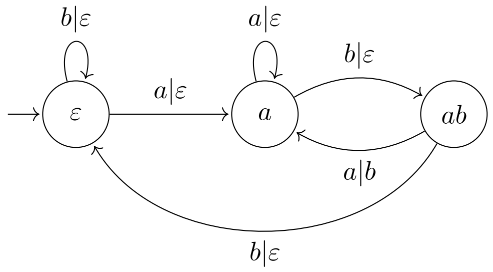

Figure 2.1: A finite state transducer realising f_{aba}.

Standard calculations, as described in Section 1.5, show that the transition monoid of each transducer realising f_w, where $|w| \leq 3$, is aperiodic. Moreover, the transition monoid of the minimal automaton recognising L is an abelian group by Theorem 2.1. Consider the following proposition.

Proposition 2.14 ([5, Proposition IX.1.1]). *Let $f : A^* \to B^*$ be a sequential function and let L be a recognisable language over B. Then, $K = Lf^{-1}$ is a recognisable language over A and the transition monoid of the minimal automaton for K divides a wreath product of the transition monoid of the minimal automaton for L by the transition monoid of the transducer realising f.*

Hence, by Proposition 2.14, the transition monoid of K divides a wreath product of an abelian group by an aperiodic monoid. Consider the following proposition.

Proposition 2.15 ([15, Theorem 7.8]). *Every language recognised by a monoid of the pseudovariety* $\mathbf{AbGrps} \rtimes \mathbf{Ap}$*, where* $\mathbf{Ap}$ *is the pseudovariety of aperiodic monoids, is of generalised star-height at most one.*

Since the pseudovarieties $\mathbf{AbGrps} \rtimes \mathbf{Ap}$ and $\mathbf{AbGrps} \wr \mathbf{Ap}$ are equal by Lemma 1.2, we conclude that K is of generalised star-height at most one.

2.5 Main result

In this section, we work in full generality and establish generalised regular expressions of generalised star-height zero and one respectively for the languages $\mathrm{Count}(w, k)$ and $\mathrm{ModCount}(w, k, n)$, where w is a word of any length.

Let w be a fixed word over an alphabet A. Define

$$B = \{b \in A^+ \mid w = bx \text{ and } w = yb \text{ for some } x, y \in A^+\},$$

the set of all proper, non-empty borders of w;

$$P = \{p \in A^+ \mid w = pb \text{ for some } b \in B\},$$

the set of prefixes of w after each border is removed as a suffix; and,

$$S = \{s \in A^+ \mid w = bs \text{ for some } b \in B\},$$

the set of suffices of w after each border is removed as a prefix.

The point of the set S is to keep track of additional occurrences of w as a subword when overlapping takes place, since every occurrence of w ends with one of its borders and this could be completed to a new occurrence of w should it be followed by an element of S. As it stands, however, the set S grants us no control over the number of further occurrences of w added as a subword. To illustrate this, consider the following example. Suppose we are interested in finding an expression for the language $\mathrm{CountBorder}(aabaabaa, k)$. Here,

$$B = \{aabaa, aa, a\} \qquad \text{and} \qquad S = \{baa, baabaa, abaabaa\}.$$

If we concatenate *aabaabaa* and *baa* then we obtain

$$aabaabaa \cdot baa = aabaabaabaa,$$

and this contains two occurrences of *aabaabaa* as a subword. However, if we concatenate *aabaabaa* and *baabaa* then we obtain

$$aabaabaa \cdot baabaa = aabaabaabaabaa,$$

and this contains three occurrences of *aabaabaa* as a subword. Thus, S gives us no control over the number of extra occurrences of *aabaabaa* that appear. In order to combat this, we restrict S to the set $\bar{S}$ as follows:

$$\bar{S} = \{s \in S \mid \nexists s' \in S \text{ such that } s = s'x \text{ for some } x \in A^+\};$$

that is, the set of suffices of w after each border is removed as a prefix satisfying the additional criterion that no element of $\bar{S}$ has another element of $\bar{S}$ as a proper prefix. The set $\bar{S}$ grants us control over counting further occurrences of subwords, as every time an element of $\bar{S}$ is concatenated on to the right of a word with suffix w we gain exactly one extra occurrence of w as a subword. This is encapsulated in the following lemma:

Lemma 2.16. *For every natural number k, we have that $w\bar{S}^k \subseteq A^*w$ and $|w\bar{S}^k|_w = k + 1$.*

Proof. Use induction on the value of k. When $k = 0$, it follows that $w\bar{S}^k = w \subseteq A^*w$ and $|w|_w = 1$. When $k = 1$, it follows that $w\bar{S}^k = w\bar{S}$. Every element s of $\bar{S}$ is a suffix of w such that $w = bs$ for some border b in B. Since w ends with any one of its borders, we can factorise w as $w = pb$ for some prefix p in P. Then

$$ws = (pb)s = p(bs) = pw \in A^*w.$$

Hence, $w\bar{S} \subseteq A^*w$. This argument shows that ws in $w\bar{S}$ contains at least two occurrences of w as a subword. Suppose that ws contains three or more occurrences of w as a subword. We know that w appears as both a prefix and a suffix of ws and that these occurrences overlap one another. Consider the following factorisation:

$$ws = w_1w_2 \ldots w_i \ldots w_{r-t}w_{r-t+1} \ldots w_rs_1s_2 \ldots s_{i-1}s_i \ldots s_t,$$

where $w = w_1w_2 \ldots w_r$ and $s = s_1s_2 \ldots s_t$. The prefix w and the suffix $w_{r-t+1} \ldots w_rs_1 \ldots s_t$ are our two known occurrences of w. Suppose that a third occurrence of w begins at w_i, where $2 \leq i \leq r-t$. Then, $w_i \ldots w_rs_1 \ldots s_{i-1} = w$.

Let $b' = w_i \dots w_r$ and $s' = s_1 \dots s_{i-1}$. By construction, b' is an element of B and s' is an element of S. However, s' is a prefix of s which contradicts the fact that s is an element of $\bar{S}$. Hence, a third occurrence of w cannot be present in ws. Therefore, $|w\bar{S}|_w = 2$, as required.

Assume that the statement holds for some natural number k. Now,

$$w\bar{S}^{k+1} = (w\bar{S}^k)\bar{S} \subseteq (A^*w)\bar{S} = A^*(w\bar{S}) \subseteq A^*(A^*w) = A^*w.$$

Finally, we know that any word in $w\bar{S}^k$ contains $k+1$ occurrences of w and has suffix w. Hence, we can factorise $w\bar{S}^{k+1}$ as $uw\bar{S}$, where uw lies in $w\bar{S}^k$. Any extra occurrences of w must be contained within the suffix $w\bar{S}$. By the base case, any word in $w\bar{S}$ contains precisely two occurrences of w. The prefix w has already been counted as an occurrence in $w\bar{S}^k$, so $w\bar{S}^{k+1}$ contains exactly one additional occurrence of w. Hence, $|w\bar{S}^{k+1}|_w = (k+1)+1$, as required. □

We now prove a partial converse of the above result. In order to do this, we need to introduce a new definition related to overlapping subwords.

Definition 2.17. Let w be a non-empty word over an alphabet A and let k be a positive integer. A word $v = a_1a_2\dots a_r$ is a *w-chain of length k* if $|v|_w = k$ and there exist indices $i_1, i_2, \dots, i_k$ such that:

(CH1) $1 = i_1 < i_2 < \cdots < i_k = r - |w| + 1$;
(CH2) $i_j + |w| > i_{j+1}$ for all $j = 1, 2, \dots, k-1$; and,
(CH3) $a_{i_j}a_{i_{j+1}} \dots a_{i_j+|w|-1} = w$ for all $j = 1, 2, \dots, k$.

As an example, every word w is a w-chain of length 1 and *abababa* is an *aba*-chain of length 3. To see this, let $x = abababa = a_1a_2a_3a_4a_5a_6a_7$. Then, the indices i_1, i_2 and i_3 take the values 1, 3 and 5 respectively. It is easiest to view the word x with its *aba*-chain highlighted, like so:

$$x = \overbrace{a \cdot b \cdot \rlap{$\underbrace{}$}a} \cdot b \cdot \overbrace{a \cdot b \cdot a}.$$

Note that the word *abababab* is neither an *abab*-chain of length 2 nor an *ab*-chain of length 4 as condition (CH2) is not satisfied – the occurrences of the subwords do not overlap.

Lemma 2.18. *If v is a w-chain of length k then v belongs to $w\bar{S}^{k-1}$.*

Proof. Use induction on the value of k. If $k = 1$ then $v = w \in w\bar{S}^0$. Assume that the statement holds for some natural number k and let v be a w-chain of length $k+1$. As such, there exist indices i_1, i_2, …, i_{k+1} satisfying conditions (CH1) to (CH3).

Let $x = a_1a_2 \dots a_{i_k}a_{i_{k+1}} \dots a_{i_k+|w|-1}$ and let $y = a_{i_k+|w|} \dots a_r$. By construction, x is a w-chain of length k and, therefore, belongs to $w\bar{S}^{k-1}$ by the induction hypothesis. Moreover, by Lemma 2.16, x has suffix w. Thus, the additional occurrence of w in v must occur as some suffix of x followed by y. The only way to add exactly one new occurrence of w is for y to belong to $\bar{S}$. Hence, $v = xy \in w\bar{S}^{k-1}\bar{S} = w\bar{S}^k$. □

Now suppose that we are searching for occurrences of w as a contiguous subword of a fixed word. A w-chain of length k is *maximal* if it is not properly contained in any w-chain of length greater than k. It is clear that every w-chain is contained within a maximal w-chain. Moreover, distinct maximal w-chains do not overlap, as if they did then they would form a w-chain of greater length, contradicting the maximality of the original chains.

Define the set of 'forbidden' words F to be the set of words that do not contain w as a subword, do not have an element of S as a prefix, do not have an element of P as a suffix or do not create an occurrence of w when sandwiched between two elements of B; that is,

$$\begin{aligned} F &= A^* \setminus (A^*wA^* \cup SA^* \cup A^*P \cup \{x \in A^* \mid w = b_1xb_2 \text{ for some } b_1, b_2 \in B\}) \\ &= (\emptyset^c w\emptyset^c \cup S\emptyset^c \cup \emptyset^c P \cup \{x \in A^* \mid w = b_1xb_2 \text{ for some } b_1, b_2 \in B\})^c. \end{aligned}$$

We will use F to separate maximal w-chains from one another in a similar fashion to how the set W was used in Lemma 2.9. The following two results highlight how F has no effect on the length of maximal w-chains.

Lemma 2.19. *$|wF|_w = 1$ and $|Fw|_w = 1$.*

Proof. Since w is a prefix of any word in wF, it follows that $|wF|_w \geq 1$. There are two possibilities for extra occurrences of w; either w occurs wholly inside F or w straddles wF. In the first case, we can write F as A^*wA^*, a form which is impossible by the definition of F. In the second case, each border of w that appears as a suffix of w has the potential to act as a prefix for another occurrence of w; that is, wF can be factorised as xbF for some x in A^* and b in B. For an extra occurrence of w to appear, the form of the word from F must be sy, where s is an element of S and y is an element of A^*. However, this form is impossible by the definition of F. Hence, $|wF|_w = 1$. A dual argument proves the other equality. □

Lemma 2.20. *$|wFw|_w = 2$.*

Proof. Since w is both a prefix and a suffix of any word in wFw, it follows that $|wFw|_w \geq 2$. By Lemma 2.19, neither wF nor Fw contain any further occurrences of w. Thus, for an extra occurrence of w to appear it must begin in the prefix w, run through F and end in the suffix w. Hence, a word in wFw can be factorised as xb_1vb_2y for some x and y in A^*, b_1 and b_2 in B and v in F. Then, $b_1vb_2 = w$, which is impossible by the definition of F. Hence, $|wFw|_w = 2$. □

Lemma 2.21. *Let A be an alphabet and let w be a non-empty word over A. Every non-empty word v over A can be uniquely decomposed as*

$$v = f_0v_1f_1v_2\dots v_jf_j,$$

where v_1, v_2, …, v_j are the maximal w-chains in v and f_0, f_1, …, f_j are words over A that contain zero occurrences of w as a subword. Moreover,

$$|v|_w = \sum_{i=1}^{j} |v_i|_w.$$

Proof. If v contains no occurrences of w as a subword then $f_0 = v$ and our decomposition is complete. Otherwise, v contains at least one occurrence of w as a subword and, therefore, contains at least one maximal w-chain. Let f_0 equal the (potentially empty) prefix of v that appears before the first occurrence of w in v. After this, the first maximal w-chain occurs; call this v_1. Now, in general, after each maximal w-chain v_i, where $1 \le i \le j-1$, there is a (potentially empty) word x containing zero occurrences of w as a subword followed by another maximal w-chain. Let $f_i = x$ and v_{i+1} equal the next maximal w-chain. Once every occurrence of w as a subword has been seen, we will have j maximal w-chains. Finally, v has a (potentially empty) suffix that contains zero occurrences of w as a subword; call this f_j. This completes the decomposition of v into maximal w-chains, as required.

Since every occurrence of w appears inside a maximal w-chain v_i, the total number of occurrences of w in v must equal the sum of the occurrences of w in each v_i. □

We are now in the position to establish a generalised regular expression for the language $\mathrm{CountBorder}(w, k)$.

Proposition 2.22. *Let A be an alphabet. For any non-empty word w over A, the language* $\mathrm{CountBorder}(w, k)$ *is represented by the generalised regular expression*

$$\bigcup_{j=1}^{k} \bigcup_{\substack{k_1,k_2,\dots,k_j \ge 0 \\ k_1+k_2+\dots+k_j = k-j}} w\bar{S}^{k_1}Fw\bar{S}^{k_2}F\dots Fw\bar{S}^{k_j}. \qquad (2.7)$$

Hence, $\mathrm{CountBorder}(w, k)$ *is of generalised star-height zero.*

Proof. Consider an arbitrary word v in $\mathrm{CountBorder}(w,k)$. By Lemma 2.21, we can decompose v as $v = f_0v_1f_1v_2\dots v_jf_j$, where v_1, v_2, …, v_j are the maximal w-chains and f_0, f_1, …, f_j are words over A that contain zero occurrences of w as a subword. Since v has border w, it must begin and end with a maximal w-chain. Hence, $f_0 = f_j = \varepsilon$. Suppose that each maximal w-chain v_i is of length $k_i + 1$. Then, by Lemma 2.18, v_i is a word of the form $w\bar{S}^{k_i}$.

We know that each f_i, where $1 \leq i \leq j-1$, contains zero occurrences of w. If f_i has a prefix from S then this would create a further occurrence of w, since v_i ends with w. This would contradict the maximality of the w-chain v_i. Hence, f_i is not a word from the set $S\emptyset^c$. Similarly, if f_i has a suffix from P then this would create a further occurrence of w, since v_{i+1} begins with w. This would contradict the maximality of the w-chain v_{i+1}. Hence, f_i is not a word from the set $\emptyset^cP$. Finally, if f_i is a word from the set

$$\{x \in A^* \mid w = b_1xb_2 \text{ for some } b_1, b_2 \in B\}$$

then a further occurrence of w would be created that would connect two distinct maximal w-chains. This contradicts the maximality of the w-chains and, therefore, cannot happen. Thus, each f_i belongs to the set F.

Now, v is a word from the set $w\bar{S}^{k_1}Fw\bar{S}^{k_2}F\dots Fw\bar{S}^{k_j}$, as required. Since each maximal w-chain is of length at least one, each of the indices k_i must be natural numbers. Moreover, as every occurrence of w appears as part of a maximal w-chain, we see that

$$k = |v|_w = \sum_{i=1}^{j} |v_i|_w = \sum_{i=1}^{j} (k_i + 1) = k_1 + k_2 + \cdots + k_j + j,$$

as required.

Conversely, let v be a word of the form given in Equation (2.7). We can factorise v as

$$v = v_1f_1v_2f_2\dots f_{j-1}v_j,$$

where each v_i is an element of $w\bar{S}^{k_i}$ and each f_i is an element of F. Lemma 2.16 shows that

$$\begin{aligned}
|v|_w &= |v_1f_1v_2f_2\dots f_{j-1}v_j|_w \\
&\geq |v_1|_w + |v_2|_w + \cdots + |v_j|_w \\
&= (k_1+1) + (k_2+1) + \cdots + (k_j+1) \\
&= k_1 + k_2 + \cdots + k_j + j \\
&= k.
\end{aligned}$$

Hence, v contains at least k occurrences of w as a subword. We show that v contains no more occurrences of w as a subword. There are four possibilities for extra occurrences of w to appear:

1. an occurrence of w straddles $v_i f_i$;
2. an occurrence of w sits wholly inside f_i;
3. an occurrence of w straddles $f_i v_{i+1}$;
4. an occurrence of w straddles $v_i f_i v_{i+1}$; that is, begins in v_i, runs through f_i and ends in v_{i+1}.

We show that in each of these cases there are no extra occurrences of w.

1. As v_i is an element of $w\bar{S}^{k_i}$, it follows by Lemma 2.16 that v_i has suffix w. Thus, if an occurrence of w straddles $v_i f_i$ then w must appear as a subword of wF at least twice. However, by Lemma 2.19, $|wF|_w = 1$. Hence, there are no extra occurrences of w.
2. An occurrence of w cannot sit wholly inside F as the definition of F discounts all words that contain w as a subword. Hence, there are no extra occurrences of w.
3. If an occurrence of w straddles $f_i v_{i+1}$ then w must appear as a subword of Fw at least twice. However, by Lemma 2.19, $|Fw|_w = 1$. Hence, there are no extra occurrences of w.
4. As v_i is an element of $w\bar{S}^{k_i}$, it follows by Lemma 2.16 that v_i has suffix w. Thus, if an occurrence of w straddles $v_i f_i v_{i+1}$ then w must appear as a subword of wFw at least three times. However, by Lemma 2.20, $|wFw|_w = 2$. Hence, there are no extra occurrences of w.

We conclude that v contains precisely k occurrences of w as a subword and that v is an element of $\mathrm{CountBorder}(w, k)$.

Finally, we note that Equation (2.7) is a generalised regular expression since both unions are finite. As the expression is star-free, it follows that $\mathrm{CountBorder}(w, k)$ is of generalised star-height zero.

□

Theorem 2.23. *Let A be an alphabet. For any non-empty word w over A and all positive integers k, the language* $\mathrm{Count}(w, k)$ *is represented by the expression*

$$(\emptyset^c w \emptyset^c \cup \emptyset^c P)^c \cdot \mathrm{CountBorder}(w, k) \cdot (S\emptyset^c \cup \emptyset^c w \emptyset^c)^c. \tag{2.8}$$

Hence, $\mathrm{Count}(w, k)$ *is of generalised star-height zero.*

Proof. Let v be an arbitrary word from $\mathrm{Count}(w, k)$. Decompose v as xyz such that the first occurrence of w is a prefix of y and the final occurrence of w is a suffix of y; that is, $y \in \text{CountBorder}(w,k)$. Consider x. By construction, every occurrence of w in v appears as a subword of y, so x cannot contain any further occurrences of w. Moreover, w is a prefix of y and, therefore, every element of B is also a prefix of y. Thus, x must not have a suffix that would create a new occurrence of w when followed by an element of B; that is, x may not have an element of P as a suffix. Hence, $x \in (\emptyset^c w \emptyset^c \cup \emptyset^c P)^c$. A dual argument shows that $z \in (S\emptyset^c \cup \emptyset^c w \emptyset^c)^c$. Hence, v can be written in the form of Equation (2.8).

Conversely, let v be a word of the form given in Equation (2.8). Decompose v as xyz, where

$$x \in (\emptyset^c w \emptyset^c \cup \emptyset^c P)^c, \quad y \in \text{CountBorder}(w,k) \quad \text{and} \quad z \in (S\emptyset^c \cup \emptyset^c w \emptyset^c)^c.$$

Since y contains precisely k occurrences of w as a subword, it follows that v contains at least k occurrences of w. Consider the prefix x. By construction, x contains no further occurrences of w as a subword. However, an extra occurrence of w could appear as a suffix of x followed by a prefix of y. The only prefixes of y that are capable of creating new occurrences of w are those that belong to the set B. In order to create a new occurrence of w using these elements of B, the suffix of x must belong to the set P and this is not allowed. Hence, the prefix x introduces no new occurrences of w. A dual argument shows that the suffix z also introduces no new occurrences of w. Hence, v contains precisely k occurrences of w as a subword and, therefore, $v \in \text{Count}(w,k)$.

Finally, we note that Equation (2.8) is a generalised regular expression. As the expression is star-free, it follows that $\text{Count}(w,k)$ is of generalised star-height zero. □

We now turn our attention to finding a generalised regular expression for the language $\text{ModCount}(w,k,n)$. In order to do this, we introduce the language $\text{Multiple}(w,n)$, which is defined by

$$\text{Multiple}(w,n) = \{v \in A^* \mid |wv|_w = n+1 \text{ and } wv \text{ has suffix } w\}.$$

We can immediately find a generalised regular expression representing the language $\text{Multiple}(w,n)$.

Lemma 2.24. *Let A be an alphabet. For any non-empty word w over A and all positive integers n, the language* $\text{Multiple}(w,n)$ *is represented by the expression*

$$\bar{S}^n \cup \bigcup_{i=0}^{n-1} \bar{S}^i F \cdot \text{CountBorder}(w, n-i). \tag{2.9}$$

Hence, $\text{Multiple}(w,n)$ *is of generalised star-height zero.*

Proof. Consider an arbitrary word $v \in \text{Multiple}(w,n)$. By definition, $|wv|_w = n+1$ and wv has suffix w. If wv is a maximal w-chain then it is, necessarily, of length $n+1$. Hence, wv is a word from the set $w\bar{S}^n$ by Lemma 2.18 and, therefore, v is a word from the set $\bar{S}^n$. Otherwise, wv is not a maximal w-chain. Let x be the longest (potentially empty) prefix of v such that wx is a maximal w-chain; let $i+1$ be its length. As wx has prefix w, $i+1 \geq 1$ and, therefore, $i \geq 0$. Similarly, $i+1 < n+1$ (that is, $i < n$) as there are still more occurrences of w to appear. Since these occurrences do not overlap with wx, we can write $wv = wxfy$, where none of the letters of f are involved in occurrences of w. This means that f must belong to the set F. Furthermore, the remaining $n-i$ occurrences of w must appear as subwords of y and y must have prefix w. Hence, y is a word in $\text{CountBorder}(w, n-i)$.

Conversely, let v be a word of the form given in Equation (2.9). If $v \in \bar{S}^n$ then $wv \in w\bar{S}^n$. By Lemma 2.16, wv has suffix w and $|wv|_w = n+1$. Hence, $v \in \text{Multiple}(w,n)$. Otherwise, v is of the form sfx, where $s \in \bar{S}^i$ for some $0 \leq i \leq n-1$, the word $f \in F$ and $x \in \text{CountBorder}(w, n-i)$. By Lemmas 2.19 and 2.20, all occurrences of w as a subword appear in s and x. Hence,

$$|wv|_w = |wsfx|_w = |ws|_w + |x|_w = (i+1) + (n-i) = n+1.$$

Moreover, by construction, wv has suffix w. Hence, $v \in \text{Multiple}(w,n)$.

Finally, we note that Equation (2.9) is a generalised regular expression since the union is finite. As the expression is star-free, it follows that $\text{Multiple}(w,n)$ is of generalised star-height zero. □

In order to find a generalised regular expression for $\text{ModCount}(w,k,n)$, we employ our general strategy by using $\text{CountBorder}(w,k)$ to count the first k occurrences of w before using $\text{Multiple}(w,n)$ to count further occurrences of w in multiples of n. We then append prefixes and suffices as appropriate, ensuring that neither creates any further occurrences of w as a subword.

Theorem 2.25. *Let A be an alphabet. For any non-empty word w over A, the language* $\text{ModCount}(w,k,n)$ *is represented by the expression*

$$(\emptyset^c w \emptyset^c \cup \emptyset^c P)^c \cdot \text{CountBorder}(w,k) \cdot \text{Multiple}(w,n)^* \cdot (S\emptyset^c \cup \emptyset^c w \emptyset^c)^c. \quad (2.10)$$

Hence, $\text{ModCount}(w,k,n)$ *is of generalised star-height at most one.*

Proof. Let $v \in \text{ModCount}(w,k,n)$. Decompose v as $uxy_1y_2\dots y_mz$ such that the first occurrence of w is a prefix of x, the kth occurrence of w is a suffix of x and the $(k+in)$th occurrence of w is a suffix of y_i for $1 \leq i \leq m$. By construction, x has border w and contains precisely k occurrences of w as a subword. Hence,

$x \in \text{CountBorder}(w, k)$. Now, each y_i has suffix w and wy_i contains precisely $n+1$ occurrences of w as a subword. Hence, $y_i \in \text{Multiple}(w, n)$ and $y_1y_2 \dots y_m \in \text{Multiple}(w, n)^*$. Finally, the prefix u cannot introduce further occurrences of w as a subword. Thus, by the same reasoning as in the proof of Theorem 2.23, $u \in (\emptyset^c w \emptyset^c \cup \emptyset^c P)^c$. A dual argument shows that $z \in (S\emptyset^c \cup \emptyset^c w \emptyset^c)$. Hence, v can be written in the form of Equation (2.10).

Conversely, let v be a word of the form given in Equation (2.10). Decompose v as $uxy_1y_2 \dots y_m z$, where $u \in (\emptyset^c w \emptyset^c \cup \emptyset^c P)^c$, the word $x \in \text{CountBorder}(w, k)$, the word $y_i \in \text{Multiple}(w, n)$ for $1 \le i \le m$ and $z \in (S\emptyset^c \cup \emptyset^c w \emptyset^c)^c$. By definition, $|x|_w = k$ and each of x, y_1, …, y_{m-1} has suffix w. Thus, each y_i, where $1 \le i \le m$, contains exactly n occurrences of w as a subword and, therefore, $|xy_1y_2 \dots y_m|_w = k + mn$. Now, by the same reasoning as in the proof of Theorem 2.23, neither the prefix u nor the suffix z introduce further occurrences of w. Hence, v contains precisely $k + mn$ occurrences of w and, therefore, $v \in \text{ModCount}(w, k, n)$.

Finally, we note that Equation (2.10) is a generalised regular expression. As the expression is of generalised star-height one, it follows that the language $\text{ModCount}(w, k, n)$ is of generalised star-height at most one. □

2.6 Examples

In this section, we present a collection of examples that showcase the generalised regular expressions found above for $\text{CountBorder}(w, k)$, $\text{Count}(w, k)$, $\text{Multiple}(w, n)$ and $\text{ModCount}(w, k, n)$.

Our first example deals with the final type of word of length three, namely aba, as seen in Section 2.4.3.

Example 2.26. In this example, the subword w under consideration is aba. We show all of the necessary steps required in order to produce explicit expressions for $\text{Count}(aba, 3)$ and $\text{ModCount}(aba, 1, 3)$. Here,

$$B = \{a\}, \qquad P = \{ab\} \qquad \text{and} \qquad S = \bar{S} = \{ba\}.$$

Though not particularly helpful or informative with regards to generalised star-height, we can write F explicitly in this case as

$$\begin{aligned} F &= (\emptyset^c aba \emptyset^c \cup ba\emptyset^c \cup \emptyset^c ab \cup b)^c \\ &= (\emptyset^c ab(a\emptyset^c \cup \varepsilon) \cup b(a\emptyset^c \cup \varepsilon))^c \\ &= ((\emptyset^c ab \cup b)(a\emptyset^c \cup \varepsilon))^c \\ &= ((\varepsilon \cup \emptyset^c a)b(a\emptyset^c \cup \varepsilon))^c. \end{aligned}$$

In order to save space, we continue to write F as opposed to the explicit expression that it represents. It follows that $\mathrm{CountBorder}(aba, k)$ is represented by the expression

$$\bigcup_{j=1}^{k} \bigcup_{\substack{k_1,k_2,\dots,k_j \geq 0 \\ k_1+k_2+\cdots+k_j=k-j}} aba(ba)^{k_1}F\dots Faba(ba)^{k_j}. \tag{2.11}$$

Though obvious, we can use Equation (2.11) to explicitly calculate that

$$\mathrm{CountBorder}(aba, 1) = aba.$$

In a similar fashion, we use Equation (2.11) to explicitly calculate that

$$\begin{aligned}\mathrm{CountBorder}(aba, 2) &= aba(ba)^1 \cup aba(ba)^0Faba(ba)^0 \\ &= ababa \cup abaFaba\end{aligned}$$

and

$$\begin{aligned}&\mathrm{CountBorder}(aba, 3) \\ &\quad = aba(ba)^2 \cup aba(ba)^1Faba(ba)^0 \cup aba(ba)^0Faba(ba)^1 \cup \\ &\qquad aba(ba)^0Faba(ba)^0Faba(ba)^0 \\ &\quad = abababa \cup ababaFaba \cup abaFababa \cup abaFabaFaba.\end{aligned}$$

Using this, we see that

$$\begin{aligned}&\mathrm{Count}(aba, 3) \\ &\quad = (\emptyset^c aba\emptyset^c \cup \emptyset^c ab)^c \cdot \mathrm{CountBorder}(aba, 3) \cdot (ba\emptyset^c \cup \emptyset^c aba\emptyset^c)^c \\ &\quad = (\emptyset^c ab(a\emptyset^c \cup \varepsilon))^c \cdot \mathrm{CountBorder}(aba, 3) \cdot ((\varepsilon \cup \emptyset^c a)ba\emptyset^c)^c.\end{aligned}$$

We now turn to finding an explicit expression for $\mathrm{ModCount}(aba, 1, 3)$. In order to do this, we must first find an explicit expression for $\mathrm{Multiple}(aba, 3)$:

$$\begin{aligned}&\mathrm{Multiple}(aba, 3) \\ &\quad = \bar{S}^3 \cup \bigcup_{i=0}^{2} \bar{S}^iF \cdot \mathrm{CountBorder}(w, 3-i) \\ &\quad = \bar{S}^3 \cup \bar{S}^2F \cdot \mathrm{CountBorder}(aba, 1) \cup \bar{S}F \cdot \mathrm{CountBorder}(aba, 2) \cup \\ &\qquad F \cdot \mathrm{CountBorder}(aba, 3) \\ &\quad = (ba)^3 \cup (ba)^2Faba \cup baF(ababa \cup abaFaba) \cup F(abaFabaFaba) \\ &\quad = bababa \cup babaFaba \cup baFababa \cup baFabaFaba \cup FabaFabaFaba\end{aligned}$$

Finally,

$$\begin{aligned}
&\text{ModCount}(aba, 1, 3) \\
&\quad = (\emptyset^c aba \emptyset^c \cup \emptyset^c ab)^c \cdot aba \cdot \text{Multiple}(aba, 3)^* \cdot (ba\emptyset^c \cup \emptyset^c aba \emptyset^c)^c \\
&\quad = (\emptyset^c ab(a\emptyset^c \cup \varepsilon))^c \cdot aba \cdot \text{Multiple}(aba, 3)^* \cdot ((\varepsilon \cup \emptyset^c a)ba\emptyset^c)^c.
\end{aligned}$$

Example 2.27. In this example, we work over the unary alphabet $A = \{a\}$ and recreate the results of Section 2.3. Here, the subword w under consideration is of the form a^r for some positive integer r. It follows that

$$B = P = S = \{a, a^2, \ldots, a^{r-1}\} \qquad \text{and} \qquad \bar{S} = \{a\}.$$

We can write F explicitly as

$$\begin{aligned}
F &= (\emptyset^c a^r \emptyset^c \cup \{a, a^2, \ldots, a^{r-1}\}a^* \cup a^*\{a, a^2, \ldots, a^{r-1}\} \cup \{\varepsilon, a, \ldots, a^{r-2}\})^c \\
&= (\emptyset^c)^c \\
&= \emptyset.
\end{aligned}$$

Now, when we substitute $F = \emptyset$ into Equation (2.7) we see that the only contribution occurs when $j = 1$. This is due to the fact that $w\emptyset = \emptyset = \emptyset w$ for all words w. Hence,

$$\text{CountBorder}(a^r, k) = \bigcup_{\substack{k_1 \geq 0 \\ k_1 = k-1}} a^{r+k_1} = a^{r+k-1}.$$

This agrees with the expression previously established in Equation (2.2).

Using this, we see that

$$\begin{aligned}
&\text{Count}(a^r, k) \\
&\quad = (\emptyset^c a^r \emptyset^c \cup \emptyset^c\{a, a^2, \ldots, a^{r-1}\})^c \cdot \text{CountBorder}(a^r, k) \cdot \\
&\qquad (\{a, a^2, \ldots, a^{r-1}\}\emptyset^c \cup \emptyset^c a^r \emptyset^c)^c \\
&\quad = (\emptyset^c a^r \emptyset^c \cup \emptyset^c a)^c \cdot \text{CountBorder}(a^r, k) \cdot (a\emptyset^c \cup \emptyset^c a^r \emptyset^c)^c \\
&\quad = (a^+)^c \cdot \text{CountBorder}(a^r, k) \cdot (a^+)^c \\
&\quad = \varepsilon \cdot \text{CountBorder}(a^r, k) \cdot \varepsilon \\
&\quad = \text{CountBorder}(a^r, k) \\
&\quad = a^{r+k-1},
\end{aligned}$$

where the third equality follows since we are working over a unary alphabet and, therefore, every non-empty word has a as both a prefix and a suffix.

Now, when we substitute $F = \emptyset$ into Equation (2.9) we see that the only contribution comes from the $\bar{S}^n$ term. Hence,

$$\text{Multiple}(a^r, n) = a^n \cup \bigcup_{i=0}^{n-1} a^i \emptyset a^{r+n-i+1} = a^n.$$

Finally,

$$\text{ModCount}(a^r, k, n) = \varepsilon \cdot a^{r+k-1} \cdot (a^n)^* \cdot \varepsilon = a^{r+k-1}(a^n)^*,$$

which agrees with the expression previously established in Equation (2.4).

Example 2.28. In this example, the subword w under consideration has ε as its only border and we aim to reproduce the results of Lemma 2.8. It follows from the various definitions that

$$B = P = S = \bar{S} = \emptyset.$$

Moreover,

$$F = (\emptyset^c w \emptyset^c \cup \emptyset\emptyset^c \cup \emptyset^c\emptyset \cup \emptyset)^c = (\emptyset \cup \emptyset^c w \emptyset^c)^c.$$

Thus, in order for a word v to belong to F it must contain no occurrences of w as a subword; that is, v is a word in Count$(w, 0)$. Now,

$$\text{CountBorder}(w, k) = \bigcup_{j=1}^{k} \bigcup_{\substack{k_1, k_2, \ldots, k_j \geq 0 \\ k_1 + k_2 + \cdots + k_j = k-j}} w\emptyset^{k_1} F \ldots F w \emptyset^{k_j},$$

and this equals the empty set unless $k_1 = k_2 = \cdots = k_j = 0$, since $\emptyset^0 = \varepsilon$. Hence,

$$\begin{aligned}\text{CountBorder}(w, k) &= w \cdot \text{Count}(w, 0) \cdot \cdots \cdot \text{Count}(w, 0) \cdot w \\ &= (w \cdot \text{Count}(w, 0))^{k-1} \cdot w.\end{aligned}$$

Using this, we see that

$$\begin{aligned}\text{Count}(w, k) &= (\emptyset^c w \emptyset^c \cup \emptyset^c\emptyset)^c \cdot \text{CountBorder}(w, k) \cdot (\emptyset\emptyset^c \cup \emptyset^c w \emptyset^c)^c \\ &= \text{Count}(w, 0) \cdot (w \cdot \text{Count}(w, 0))^{k-1} \cdot w \cdot \text{Count}(w, 0) \\ &= (\text{Count}(w, 0) \cdot w)^k \cdot \text{Count}(w, 0),\end{aligned}$$

which agrees with the expression previously established in Equation (2.5).

Now, when we substitute $\bar{S} = \emptyset$ into Equation (2.9) we see that the only contribution comes when $i = 0$. Hence,

$$\text{Multiple}(w, n) = \emptyset^n \cup \bigcup_{i=0}^{n-1} \emptyset^i \cdot \text{Count}(w, 0) \cdot \text{CountBorder}(w, n-i)$$

$$\begin{aligned}&= \emptyset \cup \varepsilon \cdot \mathrm{Count}(w,0) \cdot (w \cdot \mathrm{Count}(w,0))^{n-1} \cdot w \\ &= \emptyset \cup (\mathrm{Count}(w,0) \cdot w)^n.\end{aligned}$$

Since Multiple(w, n) is clearly non-empty, we conclude that

$$\mathrm{Multiple}(w,n) = (\mathrm{Count}(w,0) \cdot w)^n.$$

Finally,

$$\begin{aligned}&\mathrm{ModCount}(w,k,n) \\ &\quad = \mathrm{Count}(w,0) \cdot ((w \cdot \mathrm{Count}(w,0))^{k-1} \cdot w) \cdot \\ &\qquad ((\mathrm{Count}(w,0) \cdot w)^n)^* \cdot \mathrm{Count}(w,0) \\ &\quad = (\mathrm{Count}(w,0) \cdot w)^k \cdot ((\mathrm{Count}(w,0) \cdot w)^n)^* \cdot \mathrm{Count}(w,0),\end{aligned}$$

which agrees with the expression previously established in Equation (2.6).

2.7 Forming a variety of languages

In the case of scattered subwords, Thérien's Theorem (Theorem 2.1) establishes a correspondence between boolean combinations of the ScatModCount(w, k, n) languages and finite nilpotent groups. A natural question to ask is whether a similar correspondence can be constructed when we consider ModCount(w, k, n) languages instead.

By Eilenberg's Variety Theorem (Theorem 1.17), if we have a pseudovariety of monoids then we can construct a corresponding variety of monoid languages and vice versa. As such, we aim to construct a variety of monoid languages based on the ModCount(w, k, n) family of languages in the hope of finding the corresponding pseudovariety of monoids.

First, we consider the ModCount(w, k, n) languages as a family of languages in their own right. Unfortunately, this family of languages does not form a variety as condition (VL1) is violated, as illustrated by the following example.

Let $A = \{a\}$ and consider the languages

$$\mathrm{ModCount}(a,0,2) = \{\varepsilon, a^2, a^4, \dots\}$$

and

$$\mathrm{ModCount}(a,1,2) = \{a, a^3, a^5, \dots\}.$$

For condition (VL1) to be true then the union of these languages also has to be of the form ModCount(w, k, n), where w is a power of a. However,

$$\mathrm{ModCount}(a,0,2) \cup \mathrm{ModCount}(a,1,2) = A^*,$$

and A^* cannot be written in the form $\mathrm{ModCount}(w,k,n)$ for any word w that is a power of a. Hence, condition (VL1) is violated and $\mathrm{ModCount}(w,k,n)$ languages do not form a variety in their own right.

To counter this violation of condition (VL1), consider instead all boolean combinations of languages of the form $\mathrm{ModCount}(w,k,n)$ as a family of languages. Since this guarantees condition (VL1) to be true, we will consider condition (VL3) in more detail.

Let $A = \{a,b,c\}$ and $B = \{a,b\}$ be alphabets. Define a homomorphism $\varphi : A^* \to B^*$ by

$$a\varphi = a, \qquad b\varphi = b \qquad \text{and} \qquad c\varphi = \varepsilon.$$

Let $K = \mathrm{ModCount}(a^2,0,2)$ be a language over B and let

$$L = \{w \in A^* \mid w\varphi \in \mathrm{ModCount}(a^2,0,2)\} = K\varphi^{-1}.$$

We aim to show that, in this situation, L cannot be written as a boolean combination of $\mathrm{ModCount}(w,k,n)$ languages over A, which would, in turn, violate condition (VL3).

First, we note that every boolean combination of $\mathrm{ModCount}(w,k,n)$ languages can be written as a union of intersections. Indeed, if we consider a single $\mathrm{ModCount}(w,k,n)$ language, a union of $\mathrm{ModCount}(w,k,n)$ languages or an intersection of $\mathrm{ModCount}(w,k,n)$ languages then the result is trivially true. Now, we can write the complement of a ModCount language as

$$(\mathrm{ModCount}(w,k,n))^c = \bigcup_{j\in\{0,1,\dots,n-1\}\setminus\{k\}} \mathrm{ModCount}(w,j,n), \tag{2.12}$$

which is a finite union of ModCount languages. Next, consider the complement of a union of ModCount languages. By de Morgan's Laws (Equation (1.1)),

$$\left(\bigcup_{i\in I}\mathrm{ModCount}(w_i,k_i,n_i)\right)^c = \bigcap_{i\in I}(\mathrm{ModCount}(w_i,k_i,n_i))^c$$

for some index set I. We now replace each term in the intersection with its corresponding union (as established in Equation (2.12)) and use the fact that intersection distributes over union to get an expression that is a union of intersections. Finally, consider the complement of an intersection of ModCount languages. By de Morgan's Laws (Equation (1.1)),

$$\left(\bigcap_{i\in I}\mathrm{ModCount}(w_i,k_i,n_i)\right)^c = \bigcup_{i\in I}(\mathrm{ModCount}(w_i,k_i,n_i))^c$$

for some index set I. We now replace each term in the union with its corresponding union (as established in Equation (2.12)) to get a larger union of

ModCount languages. This is trivially a union of intersections. Thus, every boolean combination of $\mathrm{ModCount}(w, k, n)$ languages can be written as a union of intersections.

Suppose, for a contradiction, that L can be written as a boolean combination of ModCount languages; that is, L can be written as a union of intersections of ModCount languages like so:

$$L = \bigcup_{i \in I} \bigcap_{j \in J} \mathrm{ModCount}(w_{ij}, k_{ij}, n_{ij}),$$

where I and J are index sets. We denote the right-hand side of the equation by M. Membership of M is governed by finitely many 'counters' c_{ij}, each of which counts the occurrences of the subword w_{ij} modulo n_{ij}. Different configurations of the counters lead to words being accepted or rejected as valid words in M.

Let

$$p = \max_{\substack{i \in I \\ j \in J}} |w_{ij}|$$

and consider the word $v = a^2c^p$. We see that v is not a word in L since

$$v\varphi = (a^2c^p)\varphi = a^2$$

and this does not belong to K. However, va is a word in L since

$$(va)\varphi = (a^2c^pa)\varphi = a^3$$

and this does belong to K. Thus, the counters c_{ij} are in a reject configuration for v while for va they are in an accept configuration. Explicitly, when we concatenate v on the right with a then each of the counters for words of the form $c^l a$, where $l = 0, 1, \ldots, p - 1$, is increased by one.

If we concatenate v on the right by c^pbc^p then this increments the counters for all subwords of $c^p \cdot c^pbc^p$ whose length is at most p. If we do this again, that is, if we consider $v \cdot c^pbc^p \cdot c^pbc^p$, then exactly the same counters get incremented by precisely the same amounts. Hence, if we concatenate v on the right by $(c^pbc^p)^n$, where

$$n = \prod_{\substack{i \in I \\ j \in J}} n_{ij},$$

then all of the counters return to their original configuration; that is, the configuration of the counters for v.

Now, right concatenating $v(c^pbc^p)^n$ with a places the configuration of the counters into the same configuration as that of va. Since va is a word in M, it follows that $v(c^pbc^p)^na$ is a word in M and, hence, a word in L. However,

$(v(c^p bc^p)^n a)\varphi = a^2 b^n a$ and this does not belong to K, a contradiction. Therefore, L cannot be written as a boolean combination of languages of the form $\mathrm{ModCount}(w, k, n)$.

This violation of condition (VL3) shows that the collection of all boolean combinations of languages of the form $\mathrm{ModCount}(w, k, n)$ does not form a variety of monoid languages. Further evidence to support this statement comes from studying condition (VL2) in detail, though our results on the satisfaction of this condition are inconclusive. We consider it below.

In order to show that condition (VL2) is satisfied, we need to show that for every boolean combination of ModCount languages the corresponding left and right quotients by an arbitrary letter from the same alphabet is also contained in the boolean combination. As such, let v be a word in the language $a^{-1}\,\mathrm{ModCount}(w, k, n)$. By the definition of left quotient,

$$v \in a^{-1}\,\mathrm{ModCount}(w, k, n) \Leftrightarrow av \in \mathrm{ModCount}(w, k, n).$$

Therefore,

$$v \in \begin{cases} \mathrm{ModCount}(w, k-1, n), & \text{if } av \text{ has prefix } w \text{ and } k > 0, \\ \mathrm{ModCount}(w, n-1, n), & \text{if } av \text{ has prefix } w \text{ and } k = 0, \\ \mathrm{ModCount}(w, k, n), & \text{if } av \text{ does not have prefix } w. \end{cases}$$

Thus, the left quotient $a^{-1}\,\mathrm{ModCount}(w, k, n)$ is a proper subset of

$$\mathrm{ModCount}(w, k-1, n) \cup \mathrm{ModCount}(w, k, n) \cup \mathrm{ModCount}(w, n-1, n),$$

since some, but not all, words from each language in the union must appear in the quotient. This, however, does not rule out the fact that the left quotient $a^{-1}\,\mathrm{ModCount}(w, k, n)$ could be written as a boolean combination in a different way.

Based on the above arguments, the only remaining sensible way in which to try and form a variety from the ModCount languages is to use them as generators. In order to do this, we must begin with the ModCount languages and recursively take all boolean combinations, left and right quotients, and homomorphic preimages, hoping that the recursive process terminates. If it does then the resulting collection of languages forms the variety that we want; otherwise, such a variety does not exist.

Chapter 3

Rees Matrix Semigroups

In this chapter, we change tack and use our combinatorial results from Chapter 2 to prove new results in an algebraic setting. Specifically, we show that languages recognised by Rees zero-matrix semigroups over abelian groups and languages recognised by Rees zero-matrix semigroups over monogenic semigroups are of generalised star-height at most one.

3.1 Definitions and motivation

Let S be a semigroup, I and Λ be non-empty index sets and let P be a $|\Lambda| \times |I|$ *sandwich* matrix with entries from S. The *Rees matrix semigroup* $M[S; I, \Lambda; P]$ is the set $I \times S \times \Lambda$ equipped with the binary operation defined by

$$(i, s, \lambda)(j, t, \mu) = (i, sp_{\lambda j}t, \mu),$$

where $p_{\lambda j}$ denotes the entry of P in row λ and column j.

A related notion is that of a Rees zero-matrix semigroup. Let S be a semigroup without zero and let 0 be a new symbol not in S. Let I and Λ be non-empty index sets and let P be a $|\Lambda| \times |I|$ sandwich matrix with entries from $S \cup \{0\}$. The *Rees zero-matrix semigroup* $M^0[S; I, \Lambda; P]$ is the set $(I \times S \times \Lambda) \cup \{0\}$ equipped with the binary operation defined by

$$(i, s, \lambda)(j, t, \mu) = \begin{cases} (i, sp_{\lambda j}t, \mu) & \text{if } p_{\lambda j} \neq 0, \\ 0 & \text{if } p_{\lambda j} = 0, \end{cases}$$

and $m0 = 0 = 0m$ for all m in $M^0[S; I, \Lambda; P]$. If every row and every column of the matrix P contains a non-zero entry then P is *regular*.

Our motivation for studying languages recognised by Rees zero-matrix semigroups stems from the following theorem:

Theorem 3.1 (The Rees Theorem, [9, Theorem 3.2.3]). *If S is a semigroup with zero then it is completely 0-simple if and only if it is isomorphic to a Rees zero-matrix semigroup over a group with regular sandwich matrix.*

According to Theorem 3.1, Rees zero-matrix semigroups with finite underlying groups and regular matrices are precisely finite 0-simple semigroups. In turn, these semigroups together with zero semigroups completely exhaust principal factors of arbitrary finite semigroups. This is analogous to finite simple groups acting as the building blocks for arbitrary finite groups.

3.2 Setup

Let $M = M^0[S; I, \Lambda; P]$ be a Rees zero-matrix semigroup over a finite semigroup S, where the zero in M is denoted by 0. Let A be an alphabet and define a map $\varphi : A \to M$ by either $a\varphi = 0$ or $a\varphi = (i_a, s_a, \lambda_a)$, where a is a letter from A. Let

$$A_{(i,s,\lambda)} = (i, s, \lambda)\varphi^{-1} \qquad \text{and} \qquad A_0 = 0\varphi^{-1}.$$

Uniquely extend φ to a homomorphism $\bar{\varphi} : A^+ \to M$.

Consider the image of the word $w = a_1a_2 \dots a_r$ under $\bar{\varphi}$:

$$w\bar{\varphi} = (a_1a_2 \dots a_r)\bar{\varphi} = (a_1\bar{\varphi})(a_2\bar{\varphi}) \dots (a_r\bar{\varphi}) = (a_1\varphi)(a_2\varphi) \dots (a_r\varphi).$$

If $a_l\varphi = 0$ for at least one l in $\{1, 2, \dots, r\}$ then $w\bar{\varphi} = 0$. Otherwise, $a_l\varphi \neq 0$ for all l in $\{1, 2, \dots, r\}$ and

$$w\bar{\varphi} = \dots = (i_{a_1}, s_{a_1}, \lambda_{a_1})(i_{a_2}, s_{a_2}, \lambda_{a_2}) \dots (i_{a_r}, s_{a_r}, \lambda_{a_r}).$$

Now, if $p_{\lambda_{a_l} i_{a_{l+1}}} = 0$ for at least one l in $\{1, 2, \dots, r-1\}$ then, again, $w\bar{\varphi} = 0$. Otherwise, $p_{\lambda_{a_l} i_{a_{l+1}}} \neq 0$ for all l in $\{1, 2, \dots, r-1\}$ and

$$w\bar{\varphi} = \dots = (i_{a_1}, s_{a_1} \cdot p_{\lambda_{a_1} i_{a_2}} \cdot s_{a_2} \cdot p_{\lambda_{a_2} i_{a_3}} \cdot \dots \cdot p_{\lambda_{a_{r-1}} i_{a_r}} \cdot s_{a_r}, \lambda_{a_r}).$$

In order to find out which languages are recognised by Rees (zero-)matrix semigroups, we consider the preimages of each of the elements of M in turn; that is, we consider the preimage of the zero 0 and the preimage of an arbitrary non-zero element $m = (i, s, \lambda)$. We begin with the former case.

Proposition 3.2. *The preimage of the zero of M is of generalised star-height zero.*

Proof. According to the analysis above, a word $w = a_1a_2 \dots a_r$ belongs to the preimage of 0 if and only if at least one of the following holds:

1. a_l lies in A_0 for at least one l in $\{1, 2, \ldots, r\}$; or,
2. $p_{\lambda j} = 0$, where a_l lies in $A_{(i,s_1,\lambda)}$ and a_{l+1} lies in $A_{(j,s_2,\mu)}$.

It follows that

$$0\bar{\varphi}^{-1} = A^* A_0 A^* \cup \Big(\bigcup A^* A_{(i,s_1,\lambda)} A_{(j,s_2,\mu)} A^* \Big),$$

where the bracketed union is taken over all (i, s_1, λ) and (j, s_2, μ) in $M \backslash \{0\}$ with $p_{\lambda j} = 0$. This is a star-free regular expression by Lemma 1.24 and, therefore, $0\bar{\varphi}^{-1}$ is a language of generalised star-height zero. □

We now consider the preimage of an arbitrary non-zero element $m = (i, s, \lambda)$ of M. We begin by writing $m\bar{\varphi}^{-1}$ as the intersection of three preimages as follows:

$$m\bar{\varphi}^{-1} = (\{i\} \times S \times \Lambda)\,\bar{\varphi}^{-1} \cap (I \times \{s\} \times \Lambda)\,\bar{\varphi}^{-1} \cap (I \times S \times \{\lambda\})\,\bar{\varphi}^{-1}. \quad (3.1)$$

Note that each of these preimages fixes one of the components of m; for example, $(\{i\} \times S \times \Lambda)\bar{\varphi}^{-1}$ fixes the i component and contains all words w that get mapped to an element of M with the corresponding i component of m. Taking the intersection of these three preimages ensures that we end up with precisely those words that get mapped to m.

We study each of these preimages in turn, beginning with the first and last simultaneously.

Lemma 3.3. *In the decomposition given in Equation* (3.1)*, the preimages* $(\{i\} \times S \times \Lambda)\,\bar{\varphi}^{-1}$ *and* $(I \times S \times \{\lambda\})\,\bar{\varphi}^{-1}$ *are both regular languages of generalised star-height zero.*

Proof. As a consequence of the multiplication on M, the first letter of any word w in the preimage of $(\{i\} \times S \times \Lambda)$ must belong to the set $A_{(i,s,\mu)}$ for some s in S and some μ in Λ. This ensures that our first component is an i. The letters that follow have no influence on the first component and can, therefore, be any of the letters from A. Hence,

$$(\{i\} \times S \times \Lambda)\,\bar{\varphi}^{-1} = \left(\bigcup_{s \in S, \mu \in \Lambda} A_{(i,s,\mu)} \right) \cdot A^*.$$

Similarly, the final letter of any word w in the preimage of $(I \times S \times \{\lambda\})$ must belong to the set $A_{(j,s,\lambda)}$ for some j in I and some s in S. This ensures that our last component is a λ. The preceding letters have no influence on the final component and can, therefore, be any of the letters from A. Hence,

$$(I \times S \times \{\lambda\})\,\bar{\varphi}^{-1} = A^* \cdot \left(\bigcup_{j \in I, s \in S} A_{(j,s,\lambda)} \right).$$

In each of these expressions, the union is finite and, therefore, both expressions are regular. By Lemma 1.24, these languages both have generalised star-height zero. □

Lemma 3.3 shows us that the generalised star-height of a language recognised by M depends entirely upon the generalised star-height of the preimage $(I \times \{s\} \times \Lambda)\bar{\varphi}^{-1}$. This is because all other potential preimages are of generalised star-height zero.

In general, it is difficult to find a regular expression that represents the preimage of the set $(I \times \{s\} \times \Lambda)$. As such, we steadily work through cases where the underlying group is of an increasing nilpotency class.

3.3 Over the nilpotent group of class 0

In the simplest case, S is a nilpotent group of class 0; that is, S is the trivial group $\{1\}$. Since we have already established the preimage of 0 in Proposition 3.2, it remains to establish the preimage of $(I \times \{1\} \times \Lambda)$ under $\bar{\varphi}$.

Consider an arbitrary word $w = a_1a_2 \dots a_r$ in $(I \times \{1\} \times \Lambda)\bar{\varphi}^{-1}$. Using the notation established before Proposition 3.2, we know that $p_{\lambda_{a_l} i_{a_{l+1}}} \neq 0$ for all l in $\{1, 2, \dots, r-1\}$, and, therefore, $p_{\lambda_{a_l} i_{a_{l+1}}} = 1$ for all l in $\{1, 2, \dots, r-1\}$. Thus, w belongs to the preimage of $(I \times \{1\} \times \Lambda)$ if and only if one of the following holds:

1. $w = a$, where a lies in $A_{(i,1,\lambda)}$ for some i in I and λ in Λ; or,
2. $p_{\lambda_{a_l} i_{a_{l+1}}} = 1$ for all l in $\{1, 2, \dots, r-1\}$.

For the second option, it is tempting to write the expression

$$\bigcup_{r \geq 2} \bigcup_{\substack{p_{\lambda_l i_{l+1}} = 1 \\ l \in \{1,2,\dots,r-1\}}} A_{(i_1,1,\lambda_1)} A_{(i_2,1,\lambda_2)} \cdots A_{(i_r,1,\lambda_r)},$$

but this is not regular as the first union is infinite. Instead, we note that a condition equivalent to the second above is that w cannot contain two consecutive letters whose corresponding matrix entry is 0. Thus, it follows that an expression for $(I \times \{1\} \times \Lambda)\bar{\varphi}^{-1}$ is given by

$$\bigcup_{(i,1,\lambda) \in T \setminus \{0\}} A_{(i,1,\lambda)} \cup \left(\bigcup_{\substack{(i,1,\lambda),(j,1,\mu) \in T \setminus 0 \\ p_{\lambda j} = 0}} A^* A_{(i,1,\lambda)} A_{(j,1,\mu)} A^* \right)^c .$$

This is a star-free regular expression by Lemma 1.24. Thus, $(I \times \{1\} \times \Lambda)\bar{\varphi}^{-1}$ is a language of generalised star-height zero.

At this point, we note that a Rees zero-matrix semigroup over the trivial group with matrix entries in $\{0, 1\}$ is isomorphic to a *rectangular* 0*-band*, which is defined as follows: let I and Λ be non-empty index sets and let P be a regular matrix with entries from the set $\{0, 1\}$. Define a multiplication on the set $(I \times \Lambda) \cup \{0\}$ by

$$(i, \lambda)(j, \mu) = \begin{cases} (i, \mu) & \text{if } p_{\lambda j} = 1, \\ 0 & \text{if } p_{\lambda j} = 0, \end{cases}$$

and $s0 = 0 = 0s$ for all s in $(I, \Lambda) \cup \{0\}$. The isomorphism

$$\eta : (I \times \{1\} \times \Lambda) \cup \{0\} \to (I \times \Lambda) \cup \{0\}$$

is given by

$$(i, 1, \lambda)\eta = (i, \lambda) \quad \text{and} \quad 0\eta = 0.$$

Thus, we have the following proposition:

Proposition 3.4. *Every regular language recognised by a rectangular 0-band is of generalised star-height zero.*

Proof. Every language recognised by a rectangular 0-band can be expressed as a finite union of preimages of elements in the semigroup. Since each individual preimage is of generalised star-height zero and taking finite unions does not increase generalised star-height, the result follows. □

3.4 Over nilpotent groups of class 1

3.4.1 Cyclic groups

In this section, we explore the languages recognised by Rees zero-matrix semigroups over cyclic groups. We denote the cyclic group or order n, where n is a positive integer, by $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ and write the group operation additively.

In order to aid understanding, we make slight changes to the notation established in Section 3.2. Here, we let M denote the Rees zero-matrix semigroup $M^0[\mathbb{Z}_n; I, \Lambda; P]$ and consider the preimage of $m = (i, g, \lambda)$, where g is element of $\mathbb{Z}_n$. The index sets I and Λ and the sandwich matrix P remain as in the previous set-up. To avoid confusion, we will denote the additive identity 0 in $\mathbb{Z}_n$ by 0 and the zero of the Rees zero-matrix semigroup by $\mathbf{0}$.

At this point, we alert the reader to Section 3.4.2, where an extensive worked example aims to clarify the ideas found in the proof of the following proposition.

Proposition 3.5. *For a non-zero element $m = (i, g, \lambda)$ in M, its preimage, $m\bar{\varphi}^{-1}$, is of generalised star-height at most one.*

Proof. Consider an arbitrary word $w = a_1 a_2 \dots a_r$ in $(I \times \{g\} \times \Lambda)\,\bar{\varphi}^{-1}$. Continuing to use the notation introduced before Proposition 3.2, we know that $p_{\lambda_{a_l} i_{a_{l+1}}} \neq \mathbf{0}$ for $l = 1, 2, \dots, r-1$, and

$$g_{a_1} + p_{\lambda_{a_1} i_{a_2}} + g_{a_2} + p_{\lambda_{a_2} i_{a_3}} + \dots + p_{\lambda_{a_{r-1}} i_{a_r}} + g_{a_r} \equiv g \pmod{n}.$$

We split the above sum into two, as

$$\underbrace{g_{a_1} + g_{a_2} + \dots + g_{a_r}}_{\equiv g_1 \pmod{n}} + \underbrace{p_{\lambda_{a_1} i_{a_2}} + p_{\lambda_{a_2} i_{a_3}} + \dots + p_{\lambda_{a_{r-1}} i_{a_r}}}_{\equiv g_2 \pmod{n}} \equiv g \pmod{n},$$

and examine them separately. The first sum corresponds to the contributions from 'group' summands, while the second is the contributions from 'matrix' summands.

For the group contribution, we consider the congruence given by

$$g_{a_1} + g_{a_2} + \dots + g_{a_r} \equiv g_1 \pmod{n}.$$

Grouping together summands corresponding to the same letter, we see that the above congruence is equivalent to

$$\sum_{a \in A} g_a |w|_a \equiv g_1 \pmod{n},$$

which, in turn, is equivalent to

$$\sum_{a \in A} g_a (|w|_a \pmod{n}) \equiv g_1 \pmod{n}.$$

The point here is that while $|w|_a$ can take infinitely many values, the same is not true for $|w|_a \pmod{n}$. More formally, let U be the following set of tuples indexed by A with entries from the set $\{0, 1, \dots, n-1\}$:

$$U = \left\{ (k_a)_{a \in A} \mid \sum_{a \in A} g_a k_a \equiv g_1 \pmod{n} \right\}.$$

For any fixed tuple $(k_a)_{a \in A}$ in U, every word w such that $|w|_a \equiv k_a \pmod{n}$, where a lies in A, will have group contribution equal to $g_1 \bmod n$. The set of all such words is obtained by forming the finite intersection of the languages $\mathrm{ModCount}(a, k_a, n)$ for a in A. Taking the finite union over all tuples in U results in the expression

$$\mathrm{GrpContrib}(g_1, n) = \bigcup_{(k_a) \in U} \bigcap_{a \in A} \mathrm{ModCount}(a, k_a, n),$$

which is of generalised star-height at most one, since $\mathrm{ModCount}(a, k_a, n)$ is of generalised star-height at most one by Proposition 2.6 and Lemma 2.8.

In a similar fashion, we consider the contributions made by 'matrix' summands; that is, we consider the congruence given by

$$p_{\lambda_{a_1} i_{a_2}} + p_{\lambda_{a_2} i_{a_3}} + \cdots + p_{\lambda_{a_{r-1}} i_{a_r}} \equiv g_2 \pmod{n}.$$

Counting the contribution of each matrix entry separately, we see that the above congruence is equivalent to

$$\sum_{ab \in A^2} p_{\lambda_a i_b} |w|_{ab} \equiv g_2 \pmod{n},$$

which, in turn, is equivalent to

$$\sum_{ab \in A^2} p_{\lambda_a i_b} (|w|_{ab} \pmod{n}) \equiv g_2 \pmod{n}.$$

Consider the following finite family V of tuples indexed by A^2 with entries from the set $\{0, 1, \ldots, n-1\}$:

$$V = \left\{ (k_{ab})_{ab \in A^2} \mid \sum_{ab \in A^2} p_{\lambda_a i_b} k_{ab} \equiv g_2 \pmod{n} \right\}.$$

For a fixed tuple in V, the set of all words w satisfying $|w|_{ab} \equiv k_{ab} \pmod{n}$, where ab lies in A^2, is obtained by taking the finite intersection of the languages $\mathrm{ModCount}(ab, k_{ab}, n)$. Taking the union over all tuples in V yields

$$\mathrm{MatContrib}(g_2, n) = \bigcup_{(k_{ab}) \in V} \bigcap_{ab \in A^2} \mathrm{ModCount}(ab, k_{ab}, n),$$

which is of generalised star-height at most one, since $\mathrm{ModCount}(ab, k_{ab}, n)$ is of generalised star-height at most one by the results of Section 2.4.2.

Combining the 'group' contribution and the 'matrix' contribution appropriately leads to

$$(I \times \{g\} \times \Lambda)\, \bar{\varphi}^{-1} = \bigcup_{\substack{(g_1, g_2) \in \mathbb{Z}_n^2 \\ g_1 + g_2 \equiv g \pmod{n}}} (\mathrm{GrpContrib}(g_1, n) \cap \mathrm{MatContrib}(g_2, n)), \tag{3.2}$$

and completes the proof. □

An immediate consequence of the above proposition is the following theorem:

Theorem 3.6. *A regular language recognised by a Rees zero-matrix semigroup over a cyclic group is of generalised star-height at most one.*

Proof. Every language recognised by a Rees zero-matrix semigroup over a cyclic group can be expressed as a finite union of preimages of elements in the semigroup. Since each individual preimage is of generalised star-height at most one and taking finite unions does not increase generalised star-height, the result follows. $\square$

3.4.2 An example

We clarify some of the ideas developed in the previous section by working through an extensive example.

Example 3.7. Let $M = M[\mathbb{Z}_3; I, \Lambda; P]$ be a Rees matrix semigroup, where $I = \Lambda = \{1, 2\}$ and

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}.$$

Let $A = \{a, b, c, d\}$ and define a map $\varphi : A \to M$ by

$$a\varphi = (1, 0, 1), \quad b\varphi = (1, 1, 2), \quad c\varphi = (2, 1, 1) \quad \text{and} \quad d\varphi = (2, 2, 2).$$

Extend φ to a homomorphism $\bar{\varphi} : A^+ \to M$. Let us consider each of the relevant preimages. By Equation (3.2),

$$\begin{aligned}(I \times \{0\} \times \Lambda)\bar{\varphi}^{-1} = {} & (\mathrm{GrpContrib}(0,3) \cap \mathrm{MatContrib}(0,3)) \cup \\ & (\mathrm{GrpContrib}(1,3) \cap \mathrm{MatContrib}(2,3)) \cup \\ & (\mathrm{GrpContrib}(2,3) \cap \mathrm{MatContrib}(1,3)),\end{aligned}$$

while

$$\begin{aligned}(I \times \{1\} \times \Lambda)\bar{\varphi}^{-1} = {} & (\mathrm{GrpContrib}(0,3) \cap \mathrm{MatContrib}(1,3)) \cup \\ & (\mathrm{GrpContrib}(1,3) \cap \mathrm{MatContrib}(0,3)) \cup \\ & (\mathrm{GrpContrib}(2,3) \cap \mathrm{MatContrib}(2,3))\end{aligned}$$

and

$$\begin{aligned}(I \times \{2\} \times \Lambda)\bar{\varphi}^{-1} = {} & (\mathrm{GrpContrib}(0,3) \cap \mathrm{MatContrib}(2,3)) \cup \\ & (\mathrm{GrpContrib}(1,3) \cap \mathrm{MatContrib}(1,3)) \cup \\ & (\mathrm{GrpContrib}(2,3) \cap \mathrm{MatContrib}(0,3)).\end{aligned}$$

Thus, in order to find an explicit expression for each of the preimages

$$(I \times \{0\} \times \Lambda)\bar{\varphi}^{-1}, \quad (I \times \{1\} \times \Lambda)\bar{\varphi}^{-1} \quad \text{and} \quad (I \times \{2\} \times \Lambda)\bar{\varphi}^{-1},$$

we need to establish expressions for

$$\text{GrpContrib}(0,3), \quad \text{GrpContrib}(1,3), \quad \text{GrpContrib}(2,3),$$
$$\text{MatContrib}(0,3), \quad \text{MatContrib}(1,3) \quad \text{and} \quad \text{MatContrib}(2,3).$$

We first concentrate on the group contributions. In order to contribute $g_1 \pmod 3$ from the group elements, we must have that

$$\begin{aligned} & g_a|w|_a + g_b|w|_b + g_c|w|_c + g_d|w|_d \\ &\quad \equiv 0|w|_a + 1|w|_b + 1|w|_c + 2|w|_d \\ &\quad \equiv |w|_b + |w|_c + 2|w|_d \\ &\quad \equiv g_1 \pmod 3. \end{aligned}$$

From this we notice that the number of 'a's in a word has no effect on the group contribution while the number of 'b's, 'c's and 'd's does. For ease of notation, we denote the number of occurrences of each letter modulo n by a four-tuple (k_a, k_b, k_c, k_d), as in the proof of Proposition 3.5.

Listed below are all 81 possible four-tuples (k_a, k_b, k_c, k_d) together with the group contribution that they are responsible for:

$$\begin{aligned}
(0,0,0,0),(1,0,0,0),(2,0,0,0) &\to 0+0+2(0) \equiv 0 \pmod 3\\
(0,0,0,1),(1,0,0,1),(2,0,0,1) &\to 0+0+2(1) \equiv 2 \pmod 3\\
(0,0,0,2),(1,0,0,2),(2,0,0,2) &\to 0+0+2(2) \equiv 1 \pmod 3\\
(0,0,1,0),(1,0,1,0),(2,0,1,0) &\to 0+1+2(0) \equiv 1 \pmod 3\\
(0,0,1,1),(1,0,1,1),(2,0,1,1) &\to 0+1+2(1) \equiv 0 \pmod 3\\
(0,0,1,2),(1,0,1,2),(2,0,1,2) &\to 0+1+2(2) \equiv 2 \pmod 3\\
(0,0,2,0),(1,0,2,0),(2,0,2,0) &\to 0+2+2(0) \equiv 2 \pmod 3\\
(0,0,2,1),(1,0,2,1),(2,0,2,1) &\to 0+2+2(1) \equiv 1 \pmod 3\\
(0,0,2,2),(1,0,2,2),(2,0,2,2) &\to 0+2+2(2) \equiv 0 \pmod 3\\
(0,1,0,0),(1,1,0,0),(2,1,0,0) &\to 1+0+2(0) \equiv 1 \pmod 3\\
(0,1,0,1),(1,1,0,1),(2,1,0,1) &\to 1+0+2(1) \equiv 0 \pmod 3\\
(0,1,0,2),(1,1,0,2),(2,1,0,2) &\to 1+0+2(2) \equiv 2 \pmod 3\\
(0,1,1,0),(1,1,1,0),(2,1,1,0) &\to 1+1+2(0) \equiv 2 \pmod 3\\
(0,1,1,1),(1,1,1,1),(2,1,1,1) &\to 1+1+2(1) \equiv 1 \pmod 3\\
(0,1,1,2),(1,1,1,2),(2,1,1,2) &\to 1+1+2(2) \equiv 0 \pmod 3\\
(0,1,2,0),(1,1,2,0),(2,1,2,0) &\to 1+2+2(0) \equiv 0 \pmod 3\\
(0,1,2,1),(1,1,2,1),(2,1,2,1) &\to 1+2+2(1) \equiv 2 \pmod 3
\end{aligned}$$

$$(0,1,2,2),(1,1,2,2),(2,1,2,2) \to 1+2+2(2) \equiv 1 \pmod 3$$
$$(0,2,0,0),(1,2,0,0),(2,2,0,0) \to 2+0+2(0) \equiv 2 \pmod 3$$
$$(0,2,0,1),(1,2,0,1),(2,2,0,1) \to 2+0+2(1) \equiv 1 \pmod 3$$
$$(0,2,0,2),(1,2,0,2),(2,2,0,2) \to 2+0+2(2) \equiv 0 \pmod 3$$
$$(0,2,1,0),(1,2,1,0),(2,2,1,0) \to 2+1+2(0) \equiv 0 \pmod 3$$
$$(0,2,1,1),(1,2,1,1),(2,2,1,1) \to 2+1+2(1) \equiv 2 \pmod 3$$
$$(0,2,1,2),(1,2,1,2),(2,2,1,2) \to 2+1+2(2) \equiv 1 \pmod 3$$
$$(0,2,2,0),(1,2,2,0),(2,2,2,0) \to 2+2+2(0) \equiv 1 \pmod 3$$
$$(0,2,2,1),(1,2,2,1),(2,2,2,1) \to 2+2+2(1) \equiv 0 \pmod 3$$
$$(0,2,2,2),(1,2,2,2),(2,2,2,2) \to 2+2+2(2) \equiv 2 \pmod 3$$

As in the proof of Proposition 3.5, we gather these tuples into sets U corresponding to the group contribution that they are responsible for. For $\text{GrpContrib}(0,3)$,

$$\begin{aligned} U = \{&(0,0,0,0),(1,0,0,0),(2,0,0,0),(0,0,1,1),(1,0,1,1),(2,0,1,1),\\ &(0,0,2,2),(1,0,2,2),(2,0,2,2),(0,1,0,1),(1,1,0,1),(2,1,0,1),\\ &(0,1,1,2),(1,1,1,2),(2,1,1,2),(0,1,2,0),(1,1,2,0),(2,1,2,0),\\ &(0,2,0,2),(1,2,0,2),(2,2,0,2),(0,2,1,0),(1,2,1,0),(2,2,1,0),\\ &(0,2,2,1),(1,2,2,1),(2,2,2,1)\}. \end{aligned}$$

Similarly, for $\text{GrpContrib}(1,3)$,

$$\begin{aligned} U = \{&(0,0,0,2),(1,0,0,2),(2,0,0,2),(0,0,1,0),(1,0,1,0),(2,0,1,0),\\ &(0,0,2,1),(1,0,2,1),(2,0,2,1),(0,1,0,0),(1,1,0,0),(2,1,0,0),\\ &(0,1,1,1),(1,1,1,1),(2,1,1,1),(0,1,2,2),(1,1,2,2),(2,1,2,2),\\ &(0,2,0,1),(1,2,0,1),(2,2,0,1),(0,2,1,2),(1,2,1,2),(2,2,1,2),\\ &(0,2,2,0),(1,2,2,0),(2,2,2,0)\} \end{aligned}$$

while for $\text{GrpContrib}(2,3)$,

$$\begin{aligned} U = \{&(0,0,0,1),(1,0,0,1),(2,0,0,1),(0,0,1,2),(1,0,1,2),(2,0,1,2),\\ &(0,0,2,0),(1,0,2,0),(2,0,2,0),(0,1,0,2),(1,1,0,2),(2,1,0,2),\\ &(0,1,1,0),(1,1,1,0),(2,1,1,0),(0,1,2,1),(1,1,2,1),(2,1,2,1),\\ &(0,2,0,0),(1,2,0,0),(2,2,0,0),(0,2,1,1),(1,2,1,1),(2,2,1,1),\\ &(0,2,2,2),(1,2,2,2),(2,2,2,2)\}. \end{aligned}$$

At this point, we consider a concrete example to verify that our working is correct. Consider the word $aaacd$ over A. Then, $|w|_a = 3 \equiv 0 \pmod 3$, $|w|_b = 0$, $|w|_c = 1$ and $|w|_d = 1$. Thus, the corresponding four-tuple is $(0, 0, 1, 1)$, which means that this word has group contribution equal to $0 \pmod 3$. We can check this directly using our map $\bar{\varphi}$:

$$\begin{aligned}
(aaacd)\bar{\varphi} &= (a\bar{\varphi})(a\bar{\varphi})(a\bar{\varphi})(c\bar{\varphi})(d\bar{\varphi}) \\
&= (a\varphi)(a\varphi)(a\varphi)(c\varphi)(d\varphi) \\
&= (1,0,1)(1,0,1)(1,0,1)(2,1,1)(2,2,2) \\
&= (1, 0 + p_{11} + 0 + p_{11} + 0 + p_{12} + 1 + p_{12} + 2, 2) \\
&= (1, 0 + 2(p_{11} + p_{12}), 2).
\end{aligned}$$

In this case, our two approaches lead us to the same answer, providing evidence that our method is correct.

We have now established everything that is required to provide explicit expressions for each of the GrpContrib languages. Below, we write down the expression for $\text{GrpContrib}(0, 3)$ in detail in terms of ModCount languages, where ModCount has been abbreviated to MC due to space constraints:

$$\begin{aligned}
&\text{GrpContrib}(0, 3) \\
&\quad = \bigcup_{(k_a)\in U} \bigcap_{a\in A} \text{MC}(a, k_a, n) \\
&\quad = \bigcup_{(k_a)\in U} \text{MC}(a, k_a, 3) \cap \text{MC}(b, k_b, 3) \cap \text{MC}(c, k_c, 3) \cap \text{MC}(d, k_d, 3) \\
&\quad = (\text{MC}(a, 0, 3) \cap \text{MC}(b, 0, 3) \cap \text{MC}(c, 0, 3) \cap \text{MC}(d, 0, 3)) \cup \\
&\qquad (\text{MC}(a, 1, 3) \cap \text{MC}(b, 0, 3) \cap \text{MC}(c, 0, 3) \cap \text{MC}(d, 0, 3)) \cup \\
&\qquad (\text{MC}(a, 2, 3) \cap \text{MC}(b, 0, 3) \cap \text{MC}(c, 0, 3) \cap \text{MC}(d, 0, 3)) \cup \\
&\qquad (\text{MC}(a, 0, 3) \cap \text{MC}(b, 0, 3) \cap \text{MC}(c, 1, 3) \cap \text{MC}(d, 1, 3)) \cup \\
&\qquad (\text{MC}(a, 1, 3) \cap \text{MC}(b, 0, 3) \cap \text{MC}(c, 1, 3) \cap \text{MC}(d, 1, 3)) \cup \\
&\qquad (\text{MC}(a, 2, 3) \cap \text{MC}(b, 0, 3) \cap \text{MC}(c, 1, 3) \cap \text{MC}(d, 1, 3)) \cup \\
&\qquad (\text{MC}(a, 0, 3) \cap \text{MC}(b, 0, 3) \cap \text{MC}(c, 2, 3) \cap \text{MC}(d, 2, 3)) \cup \\
&\qquad (\text{MC}(a, 1, 3) \cap \text{MC}(b, 0, 3) \cap \text{MC}(c, 2, 3) \cap \text{MC}(d, 2, 3)) \cup \\
&\qquad (\text{MC}(a, 2, 3) \cap \text{MC}(b, 0, 3) \cap \text{MC}(c, 2, 3) \cap \text{MC}(d, 2, 3)) \cup \\
&\qquad (\text{MC}(a, 0, 3) \cap \text{MC}(b, 1, 3) \cap \text{MC}(c, 0, 3) \cap \text{MC}(d, 1, 3)) \cup \\
&\qquad (\text{MC}(a, 1, 3) \cap \text{MC}(b, 1, 3) \cap \text{MC}(c, 0, 3) \cap \text{MC}(d, 1, 3)) \cup \\
&\qquad (\text{MC}(a, 2, 3) \cap \text{MC}(b, 1, 3) \cap \text{MC}(c, 0, 3) \cap \text{MC}(d, 1, 3)) \cup \\
&\qquad (\text{MC}(a, 0, 3) \cap \text{MC}(b, 1, 3) \cap \text{MC}(c, 1, 3) \cap \text{MC}(d, 2, 3)) \cup
\end{aligned}$$

$$\begin{aligned}
&(\mathrm{MC}(a,1,3)\cap \mathrm{MC}(b,1,3)\cap \mathrm{MC}(c,1,3)\cap \mathrm{MC}(d,2,3))\cup\\
&(\mathrm{MC}(a,2,3)\cap \mathrm{MC}(b,1,3)\cap \mathrm{MC}(c,1,3)\cap \mathrm{MC}(d,2,3))\cup\\
&(\mathrm{MC}(a,0,3)\cap \mathrm{MC}(b,1,3)\cap \mathrm{MC}(c,2,3)\cap \mathrm{MC}(d,0,3))\cup\\
&(\mathrm{MC}(a,1,3)\cap \mathrm{MC}(b,1,3)\cap \mathrm{MC}(c,2,3)\cap \mathrm{MC}(d,0,3))\cup\\
&(\mathrm{MC}(a,2,3)\cap \mathrm{MC}(b,1,3)\cap \mathrm{MC}(c,2,3)\cap \mathrm{MC}(d,0,3))\cup\\
&(\mathrm{MC}(a,0,3)\cap \mathrm{MC}(b,2,3)\cap \mathrm{MC}(c,0,3)\cap \mathrm{MC}(d,2,3))\cup\\
&(\mathrm{MC}(a,1,3)\cap \mathrm{MC}(b,2,3)\cap \mathrm{MC}(c,0,3)\cap \mathrm{MC}(d,2,3))\cup\\
&(\mathrm{MC}(a,2,3)\cap \mathrm{MC}(b,2,3)\cap \mathrm{MC}(c,0,3)\cap \mathrm{MC}(d,2,3))\cup\\
&(\mathrm{MC}(a,0,3)\cap \mathrm{MC}(b,2,3)\cap \mathrm{MC}(c,1,3)\cap \mathrm{MC}(d,0,3))\cup\\
&(\mathrm{MC}(a,1,3)\cap \mathrm{MC}(b,2,3)\cap \mathrm{MC}(c,1,3)\cap \mathrm{MC}(d,0,3))\cup\\
&(\mathrm{MC}(a,2,3)\cap \mathrm{MC}(b,2,3)\cap \mathrm{MC}(c,1,3)\cap \mathrm{MC}(d,0,3))\cup\\
&(\mathrm{MC}(a,0,3)\cap \mathrm{MC}(b,2,3)\cap \mathrm{MC}(c,2,3)\cap \mathrm{MC}(d,1,3))\cup\\
&(\mathrm{MC}(a,1,3)\cap \mathrm{MC}(b,2,3)\cap \mathrm{MC}(c,2,3)\cap \mathrm{MC}(d,1,3))\cup\\
&(\mathrm{MC}(a,2,3)\cap \mathrm{MC}(b,2,3)\cap \mathrm{MC}(c,2,3)\cap \mathrm{MC}(d,1,3)).
\end{aligned}$$

Explicit expressions for GrpContrib$(1,3)$ and GrpContrib$(2,3)$ can also be written down at this point, but we do not do this here.

We now turn our attention to the contributions made by the matrix entries. In order to contribute $g_2 \pmod 3$ from the matrix entries, we must have that

$$\begin{aligned}
&p_{\lambda_a i_a}|w|_{aa} + p_{\lambda_a i_b}|w|_{ab} + p_{\lambda_a i_c}|w|_{ac} + \cdots + p_{\lambda_d i_c}|w|_{dc} + p_{\lambda_d i_d}|w|_{dd}\\
&\quad = p_{11}|w|_{aa} + p_{11}|w|_{ab} + p_{12}|w|_{ac} + \cdots + p_{22}|w|_{dc} + p_{22}|w|_{dd}\\
&\quad = p_{11}(|w|_{aa} + |w|_{ab} + |w|_{ca} + |w|_{cb}) + p_{12}(|w|_{ac} + |w|_{ad} + |w|_{cc} + |w|_{cd}) +\\
&\quad\quad p_{21}(|w|_{ba} + |w|_{bb} + |w|_{da} + |w|_{db}) + p_{22}(|w|_{bc} + |w|_{bd} + |w|_{dc} + |w|_{dd})\\
&\quad = 0(|w|_{aa} + |w|_{ab} + |w|_{ca} + |w|_{cb}) + 1(|w|_{ac} + |w|_{ad} + |w|_{cc} + |w|_{cd}) +\\
&\quad\quad 1(|w|_{ba} + |w|_{bb} + |w|_{da} + |w|_{db}) + 2(|w|_{bc} + |w|_{bd} + |w|_{dc} + |w|_{dd})\\
&\quad = |w|_{ac} + |w|_{ad} + |w|_{ba} + |w|_{bb} + |w|_{cc} + |w|_{cd} +\\
&\quad\quad |w|_{da} + |w|_{db} + 2(|w|_{bc} + |w|_{bd} + |w|_{dc} + |w|_{dd})\\
&\quad \equiv g_2 \pmod 3.
\end{aligned}$$

From this we notice that the number of occurrences of aa, ab, ca and cb as a contiguous subword has no effect on the matrix contribution while the number of occurrences of all other subwords of length two does. According to the proof of Proposition 3.5, we should denote the number of occurrences of each subword of length two modulo 3 by a 16-tuple, like so:

$$(k_{aa}, k_{ab}, k_{ac}, k_{ad}, k_{ba}, k_{bb}, k_{bc}, k_{bd}, k_{ca}, k_{cb}, k_{cc}, k_{cd}, k_{da}, k_{db}, k_{dc}, k_{dd})$$

Unlike in the group contribution case, it is infeasible to list all $3^{16} = 43046721$ tuples. Therefore, we will consider a small handful of words to check that our method does, in fact, work as anticipated.

First, we consider the word $aaacd$ over A, as seen in the group contribution case. Here, $|w|_{aa} = 2$, $|w|_{ac} = 1$, $|w|_{cd} = 1$ and all other subwords of length two do not occur. Thus, the corresponding 16-tuple is

$$(2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0),$$

which means that this word has matrix contribution equal to

$$1 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 2(0 + 0 + 0 + 0) \equiv 2 \pmod 3.$$

We can check this directly using our map $\bar{\varphi}$:

$$(aaacd)\bar{\varphi} = \cdots = (1, 0 + 2(p_{11} + p_{12}), 2) = (1, 0 + 2(0 + 1), 2) = (1, 0 + 2, 2).$$

In both cases, we see that the matrix contribution is 2. Combined with the group contribution of 0, we see that the word $aaacd$ lies in the preimage of $(I \times \{2\} \times \Lambda)$; specifically, it lies in $(1, 2, 2)\bar{\varphi}^{-1}$.

Now consider the word $abacddcaba$ over A. Here,

$$\begin{aligned}
&|w|_{aa} = 0, \quad |w|_{ab} = 2, \quad |w|_{ac} = 1, \quad |w|_{ad} = 0,\\
&|w|_{ba} = 2, \quad |w|_{bb} = 0, \quad |w|_{bc} = 0, \quad |w|_{bd} = 0,\\
&|w|_{ca} = 1, \quad |w|_{cb} = 0, \quad |w|_{cc} = 0, \quad |w|_{cd} = 1,\\
&|w|_{da} = 0, \quad |w|_{db} = 0, \quad |w|_{dc} = 1, \quad \text{and} \quad |w|_{dd} = 1.
\end{aligned}$$

Thus, the corresponding 16-tuple is

$$(0, 2, 1, 0, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1),$$

which means that this word has matrix contribution equal to

$$1 + 0 + 2 + 0 + 0 + 1 + 0 + 0 + 2(0 + 0 + 1 + 1) = 8 \equiv 2 \pmod 3.$$

We can check this directly using our map $\bar{\varphi}$:

$$\begin{aligned}
(abacddcaba)\bar{\varphi} &= (a\bar{\varphi})(b\bar{\varphi})(a\bar{\varphi}) \dots (b\bar{\varphi})(a\bar{\varphi})\\
&= (a\varphi)(b\varphi)(a\varphi) \dots (b\varphi)(a\varphi)\\
&= (1, 0, 1)(1, 1, 2)(1, 0, 1) \dots (1, 1, 2)(1, 0, 1)\\
&= (1, 0 + p_{11} + 1 + p_{21} + 0 + \cdots + 1 + p_{21} + 0, 1)\\
&= (1, 2 + (3p_{11} + 2p_{12} + 2p_{21} + 2p_{22}), 1)\\
&= (1, 2 + 8, 1)
\end{aligned}$$

$$= (1, 2+2, 1).$$

In both cases, we see that the matrix contribution is 2. Combined with the group contribution of 2 (as established in the latter calculation), we see that the word *abacddcaba* lies in the preimage of $(I \times \{1\} \times \Lambda)$; specifically, it lies in $(1,1,1)\bar{\varphi}^{-1}$.

3.4.3 Direct products

In order to extend Theorem 3.6 to Rees zero-matrix semigroups over abelian groups, we make use of properties of homomorphisms and projection maps and appeal to the Fundamental Theorem of Finite Abelian Groups, as stated in Theorem 1.1.

We begin with some general theory concerning Rees zero-matrix semigroups over direct products of semigroups. Consider a Rees zero-matrix semigroup $M^0[S \times T; I, \Lambda; R]$, with $R = (r_{\lambda i})$, where $r_{\lambda i} = (p_{\lambda i}, q_{\lambda i})$ lies in $S \times T$ or $r_{\lambda i} = 0_{S\times T}$, the zero element. Define two further Rees matrix semigroups $M^0[S; I, \Lambda; P]$ and $M^0[T; I, \Lambda; Q]$, with zeros 0_S and 0_T respectively, and matrices P and Q defined by $P = (p_{\lambda i})$ and $Q = (q_{\lambda i})$, where we take $p_{\lambda i} = 0_S$ and $q_{\lambda i} = 0_T$ whenever $r_{\lambda i} = 0_{S\times T}$.

Define the natural projections

$$\pi_S : M^0[S \times T; I, \Lambda; R] \to M^0[S; I, \Lambda; P], \text{ and}$$
$$\pi_T : M^0[S \times T; I, \Lambda; R] \to M^0[T; I, \Lambda; Q]$$

by

$$(i, (s,t), \lambda)\pi_S = (i, s, \lambda), \quad (0_{S\times T})\pi_S = 0_S,$$
$$(i, (s,t), \lambda)\pi_T = (i, t, \lambda), \quad \text{and} \quad (0_{S\times T})\pi_T = 0_T.$$

Note that these projections are homomorphisms. Indeed, let $(i_1, (s_1, t_1), \lambda_1)$ and $(i_2, (s_2, t_2), \lambda_2)$ be non-zero elements of $M^0[S \times T; I, \Lambda; R]$. Then

$$\begin{aligned}
&((i_1, (s_1,t_1), \lambda_1)(i_2, (s_2,t_2), \lambda_2))\pi_S \\
&\quad = (i_1, (s_1,t_1) r_{\lambda_1 i_2} (s_2,t_2), \lambda_2)\pi_S \\
&\quad = (i_1, (s_1,t_1)(p_{\lambda_1 i_2}, q_{\lambda_1 i_2})(s_2,t_2), \lambda_2)\pi_S \\
&\quad = (i_1, (s_1 p_{\lambda_1 i_2} s_2, t_1 q_{\lambda_1 i_2} t_2), \lambda_2)\pi_S \\
&\quad = (i_1, s_1 p_{\lambda_1 i_2} s_2, \lambda_2)
\end{aligned}$$

and

$$((i_1, (s_1,t_1), \lambda_1)\pi_S)((i_2, (s_2,t_2), \lambda_2)\pi_S)$$

$$
\begin{aligned}
&= (i_1, s_1, \lambda_1)(i_2, s_2, \lambda_2)\\
&= (i_1, s_1 p_{\lambda_1 i_2} s_2, \lambda_2).
\end{aligned}
$$

Hence, π_S is a homomorphism. A similar argument shows that π_T is also a homomorphism.

Now suppose that we are given an alphabet A and a map $\varphi : A \to M^0[S \times T; I, \Lambda; R]$, which extends uniquely to a homomorphism $\bar{\varphi} : A^+ \to M^0[S \times T; I, \Lambda; R]$. Then the compositions $\bar{\varphi}\pi_S$ and $\bar{\varphi}\pi_T$ are homomorphisms from A^+ to $M^0[S; I, \Lambda; P]$ and $M^0[T; I, \Lambda; Q]$ respectively. The entire setup is summarised in Figure 3.1.

$$\begin{tikzcd}
A \arrow[rr, hook, "\iota"] \arrow[dr, "\varphi"'] & & A^+ \arrow[dl, "\bar{\varphi}"] \\
& M^0[S \times T; I, \Lambda; R] \arrow[dl, two heads, "\pi_S"'] \arrow[dr, two heads, "\pi_T"] & \\
M^0[S; I, \Lambda; P] & & M^0[T; I, \Lambda; Q]
\end{tikzcd}$$

Figure 3.1: Commutative diagram for the setup in Section 3.4.3.

In the following lemma we relate the preimage of a non-zero element in $M^0[S \times T; I, \Lambda; R]$ to the preimages of non-zero elements in $M^0[S; I, \Lambda; P]$ and $M^0[T; I, \Lambda; Q]$. Similarly, we relate the preimage of $0_{S\times T}$ to the preimages of 0_S and 0_T.

Lemma 3.8. *For any $(i, (s, t), \lambda)$ in $M^0[S \times T; I, \Lambda; R]$,*

$$(i, (s, t), \lambda)\,\bar{\varphi}^{-1} = (i, s, \lambda)(\bar{\varphi}\pi_S)^{-1} \cap (i, t, \lambda)(\bar{\varphi}\pi_T)^{-1}.$$

Similarly,

$$0_{S\times T}\bar{\varphi}^{-1} = 0_S(\bar{\varphi}\pi_S)^{-1} \cap 0_T(\bar{\varphi}\pi_T)^{-1}.$$

Proof. First, consider the former equality and suppose that w is an element of $(i, (s, t), \lambda)\,\bar{\varphi}^{-1}$; that is, $w\bar{\varphi} = (i, (s, t), \lambda)$. Then

$$w(\bar{\varphi}\pi_S) = (w\bar{\varphi})\pi_S = (i, (s, t), \lambda)\,\pi_S = (i, s, \lambda),$$

and

$$w(\bar{\varphi}\pi_T) = (w\bar{\varphi})\pi_T = (i, (s, t), \lambda)\,\pi_T = (i, t, \lambda).$$

Hence, w lies in $(i, s, \lambda)(\bar{\varphi}\pi_S)^{-1}$ and w lies in $(i, t, \lambda)(\bar{\varphi}\pi_T)^{-1}$, and, therefore, w must lie in their intersection.

Conversely, suppose that

$$w \in (i, s, \lambda)(\bar{\varphi}\pi_S)^{-1} \cap (i, t, \lambda)(\bar{\varphi}\pi_T)^{-1},$$

so that

$$w(\bar{\varphi}\pi_S) = (i, s, \lambda) \quad \text{and} \quad w(\bar{\varphi}\pi_T) = (i, t, \lambda).$$

Note that $w\bar{\varphi} \neq 0_{S\times T}$, since $w\bar{\varphi}$ lies in the preimage of (i, s, λ) under π_S whereas $0_{S\times T}$ lies in the preimage of 0_S under π_S. Thus, $w\bar{\varphi} = (i_w, (s_w, t_w), \lambda_w)$ for some

$$i_w \in I, \qquad (s_w, t_w) \in S \times T \qquad \text{and} \qquad \lambda_w \in \Lambda.$$

Now,

$$(i, s, \lambda) = (w\bar{\varphi})\pi_S = (i_w, (s_w, t_w), \lambda_w)\, \pi_S = (i_w, s_w, \lambda_w)$$

and, similarly,

$$(i, t, \lambda) = (w\bar{\varphi})\pi_T = (i_w, (s_w, t_w), \lambda_w)\, \pi_T = (i_w, t_w, \lambda_w).$$

Hence,

$$i_w = i, \qquad s_w = s, \qquad t_w = t \qquad \text{and} \qquad \lambda_w = \lambda.$$

Therefore,

$$w\bar{\varphi} = (i_w, (s_w, t_w), \lambda_w) = (i, (s, t), \lambda)$$

and w is an element of $(i, (s, t), \lambda)\bar{\varphi}^{-1}$.

Now consider the second equality and let w be an element of $0_{S\times T}\bar{\varphi}^{-1}$; that is, $w\bar{\varphi} = 0_{S\times T}$. Then

$$w(\bar{\varphi}\pi_S) = (w\bar{\varphi})\pi_S = 0_{S\times T}\pi_S = 0_S$$

and

$$w(\bar{\varphi}\pi_T) = (w\bar{\varphi})\pi_T = 0_{S\times T}\pi_T = 0_T.$$

Hence, w lies in $0_S(\bar{\varphi}\pi_S)^{-1}$ and w lies in $0_T(\bar{\varphi}\pi_T)^{-1}$, and, therefore, w must lie in their intersection.

Conversely, suppose that

$$w \in 0_S(\bar{\varphi}\pi_S)^{-1} \cap 0_T(\bar{\varphi}\pi_T)^{-1},$$

so that

$$(w\bar{\varphi})\pi_S = w(\bar{\varphi}\pi_S) = 0_S \quad \text{and} \quad (w\bar{\varphi})\pi_T = w(\bar{\varphi}\pi_T) = 0_T.$$

Now, $w\bar{\varphi}$ lies in $0_S\pi_S^{-1} = \{0_{S\times T}\}$. Hence, $w\bar{\varphi} = 0_{S\times T}$ and w is an element of $0_{S\times T}\bar{\varphi}^{-1}$. $\square$

We can now prove the following:

Theorem 3.9. *Let S and T be finite semigroups. If languages recognised by finite Rees zero-matrix semigroups over S or T all have generalised star-height at most h, then all the languages recognised by finite Rees zero-matrix semigroups over the direct product $S \times T$ also have generalised star-height at most h.*

Proof. Lemma 3.8 allows us to express the preimage of an element in the Rees zero-matrix semigroup over the direct product as the intersection of two preimages of elements in Rees zero-matrix semigroups over the factors. Since the preimage of any subset is a finite union of preimages of elements, the result follows. □

3.4.4 Extending to abelian groups

By combining the above results we can now extend Theorem 3.6 to Rees zero-matrix semigroups over abelian groups.

Theorem 3.10. *A regular language recognised by a Rees zero-matrix semigroup over an abelian group is of generalised star-height at most one.*

Proof. Invoking the Fundamental Theorem of Finite Abelian Groups (Theorem 1.1) and applying Theorem 3.9 a finite number of times to Rees zero-matrix semigroups over cyclic groups yields the result. □

Theorem 3.10 can also be deduced from existing theoretical results when attention is restricted to the basic Rees matrix construction (without zero). Indeed, let M be a Rees matrix semigroup over an abelian group G. Consider the following proposition.

Proposition 3.11 ([5, Proposition XI.3.1]). *Let M be a Rees matrix semigroup over a semigroup S. Then, M divides a wreath product of S by an aperiodic monoid.*

It follows, by Corollary 1.15, that if a language is recognised by M then it is also recognised by an element of $\mathbf{AbGrps} \wr \mathbf{Ap}$. Since the pseudovarieties $\mathbf{AbGrps} \wr \mathbf{Ap}$ and $\mathbf{AbGrps} \rtimes \mathbf{Ap}$ are equal by Lemma 1.2 and every language recognised by a monoid of the pseudovariety $\mathbf{AbGrps} \rtimes \mathbf{Ap}$ is of generalised star-height at most one (Proposition 2.15), it follows that every language recognised by M is of generalised star-height at most one.

3.5 Over monogenic semigroups

Let S be a finite *monogenic semigroup* with *index* l and *period* q; that is,

$$S = \langle 1 \rangle = \{1, 2, \ldots, l, l+1, \ldots, l+q-1\},$$

where l and q are chosen minimally such that $l + q = l$. The subset

$$K = \{l, l+1, \ldots, l+q-1\}$$

of S forms a cyclic group of order q. Due to this connection between monogenic semigroups and cyclic groups, we should be able to analyse the generalised star-height of Rees zero-matrix semigroups over monogenic semigroups in a similar way to that of Rees zero-matrix semigroups over cyclic groups in Section 3.4.1.

Since we have already established the preimage of 0 in Proposition 3.2, it remains to establish the preimage of $(I \times \{s\} \times \Lambda)$ under $\bar{\varphi}^{-1}$. We split into the following two cases:

(Case 1) $1 \leq s < l$; and,
(Case 2) $l \leq s \leq l+q-1$.

In Case 1, we mirror the proof of Proposition 3.5 noting that we are only interested in exact counting as opposed to modular counting. As such, we consider an arbitrary word $w = a_1 a_2 \ldots a_r$ in $(I \times \{s\} \times \Lambda)\bar{\varphi}^{-1}$. Continuing to use the notation introduced before Proposition 3.2, we know that $p_{\lambda_{a_j} i_{j+1}} \neq \mathbf{0}$ for $j = 1, 2, \ldots, r-1$, and

$$s_{a_1} + p_{\lambda_{a_1} i_{a_2}} + s_{a_2} + p_{\lambda_{a_2} i_{a_3}} + \cdots + p_{\lambda_{a_{r-1}} i_{a_r}} + s_{a_r} = s.$$

We split the above sum into two, as

$$\underbrace{s_{a_1} + s_{a_2} + \cdots + s_{a_r}}_{=s_1} + \underbrace{p_{\lambda_{a_1} i_{a_2}} + p_{\lambda_{a_2} i_{a_3}} + \cdots + p_{\lambda_{a_{r-1}} i_{a_r}}}_{=s_2} = s,$$

and examine them separately. The first sum corresponds to the contributions from 'group' summands, while the second is the contributions from 'matrix' summands. Notice that the value of s_1 is always at least 1 since every s_{a_j} is at least 1. For s_2, the same is true except when $r = 1$ as in this case there is no contribution from matrix elements. In this case, $s_2 = 0$.

For the group contribution, we consider the equation

$$s_1 = s_{a_1} + s_{a_2} + \cdots + s_{a_r} = \sum_{a \in A} s_a |w|_a.$$

Let U be the following set of tuples indexed by A containing entries from the set $\{0, 1, 2, \ldots, l-1\}$:

$$U = \left\{ (k_a)_{a \in A} \mid \sum_{a \in A} s_a k_a = s_1 \right\}.$$

As in the proof of Proposition 3.5, it follows that

$$\mathrm{GrpContrib}(s_1) = \bigcup_{(k_a)\in U} \bigcap_{a\in A} \mathrm{Count}(a, k_a).$$

Similarly, for the matrix contribution, we consider the equation

$$s_2 = p_{\lambda_{a_1} i_{a_2}} + p_{\lambda_{a_2} i_{a_3}} + \cdots + p_{\lambda_{a_{r-1}} i_{a_r}} = \sum_{ab\in A^2} p_{\lambda_a i_b} |w|_{ab}.$$

Let V be the following set of tuples indexed by A^2 containing entries from the set $\{0, 1, 2, \ldots, l-1\}$:

$$V = \left\{ (k_{ab})_{ab\in A^2} \mid \sum_{ab\in A^2} p_{\lambda_a i_b} k_{ab} = s_2 \right\}.$$

As in the proof of Proposition 3.5, it follows that

$$\mathrm{MatContrib}(s_2) = \bigcup_{(k_{ab})\in V} \bigcap_{ab\in A^2} \mathrm{Count}(ab, k_{ab}).$$

Combining the 'group' contribution and the 'matrix' contribution appropriately leads to a regular expression for $(I \times \{s\} \times \Lambda)\bar{\varphi}^{-1}$, given by

$$\bigcup_{i\in I, \lambda\in\Lambda} A_{(i,s,\lambda)} \cup \bigcup_{\substack{(s_1,s_2)\in\{1,2,\ldots,s-1\}^2 \\ s_1+s_2=s}} (\mathrm{GrpContrib}(s_1) \cap \mathrm{MatContrib}(s_2)).$$

The first union corresponds to all of the letters of A that have s as their group contribution. This is required as a separate term as these letters have no matrix contribution. It follows that the preimage $(I \times \{s\} \times \Lambda)\bar{\varphi}^{-1}$ is of generalised star-height zero whenever s is less than l.

Now, we consider Case 2. Again, consider an arbitrary word $w = a_1 a_2 \ldots a_r$ in $(I \times \{s\} \times \Lambda)\bar{\varphi}^{-1}$ where, in this situation, s is greater than or equal to l. Continuing to use the notation introduced before Proposition 3.2, we know that $p_{\lambda_{a_j} i_{j+1}} \neq \mathbf{0}$ for $j = 1, 2, \ldots, r-1$, and, therefore,

$$s_{a_1} + p_{\lambda_{a_1} i_{a_2}} + s_{a_2} + p_{\lambda_{a_2} i_{a_3}} + \cdots + p_{\lambda_{a_{r-1}} i_{a_r}} + s_{a_r} \geq l.$$

As such, w does not belong to any of the preimages of a semigroup element less than l; that is, w belongs to the set

$$X = A^* \setminus \bigcup_{j=1}^{l-1} (I \times \{j\} \times \Lambda)\bar{\varphi}^{-1}$$

From Case 1, each of the preimages $(I \times \{j\} \times \Lambda)\bar{\varphi}^{-1}$, where $1 \leq j \leq l-1$, is of generalised star-height zero. Since the union is finite, it follows that X is a regular language of generalised star-height zero.

Now, since

$$l \leq s \leq l + q - 1,$$

it follows that

$$0 \leq s - l \leq q - 1.$$

Thus, $s - l$ is an element of the cyclic group $\mathbb{Z}_q$. Therefore, we consider

$$s_{a_1} + p_{\lambda_{a_1} i_{a_2}} + s_{a_2} + p_{\lambda_{a_2} i_{a_3}} + \cdots + p_{\lambda_{a_{r-1}} i_{a_r}} + s_{a_r} \equiv (s - l) \pmod{q}.$$

We can now apply the proof of Proposition 3.5 directly. Intersecting the resulting expression with the language X ensures that we capture only those words whose contribution has passed the threshold for entering the cyclic group; namely, the index l. It follows that the preimage $(I \times \{s\} \times \Lambda)\bar{\varphi}^{-1}$ is of generalised star-height at most one whenever $l \leq s \leq l + q - 1$.

The following theorem is a near-immediate consequence of our foregoing results:

Theorem 3.12. *A regular language recognised by a Rees zero-matrix semigroup over a finite monogenic semigroup is of generalised star-height at most one.*

Proof. Every language recognised by a Rees zero-matrix semigroup over a finite monogenic semigroup can be expressed as a finite union of preimages of elements in the semigroup. Since each individual preimage is of generalised star-height at most one and taking finite unions does not increase generalised star-height, the result follows. $\square$

Chapter 4

Counting Arrows

4.1 Definitions and motivation

In this chapter, we turn our attention to languages recognised by finite groups of a given order. By Theorem 2.2, we conclude that every language recognised by a finite group of order p or p^2, where p is a prime number, is of generalised star-height at most one, since groups of these orders are abelian. This proves a substantial amount of the following theorem, which acts as our motivation in this chapter.

Theorem 4.1 ([15, Corollary 7.7])**.** *Every language recognised by a finite group of order less than* 12 *is of generalised star-height at most one.*

In Table 4.1, we list all finite groups of order less than 12 alongside the result used to prove that a language recognised by that group is of generalised star-height at most one. We see that only 4 of the 19 groups are not abelian; namely, Dih_3, Dih_4, Q_8 and Dih_5. Now, Dih_4 and Q_8 are both nilpotent of class 2, so languages recognised by both of these groups are of generalised star-height at most one by Theorem 2.3.

It remains for us to determine the generalised star-height of the languages recognised by the two dihedral groups Dih_3 and Dih_5. The group Dih_3 can be decomposed as the semidirect product $\mathbb{Z}_3 \rtimes \mathbb{Z}_2$ and, similarly, Dih_5 can be decomposed as the semidirect product $\mathbb{Z}_5 \rtimes \mathbb{Z}_2$. Both of these decompositions are semidirect products of an abelian group by the cyclic group $\mathbb{Z}_2$. Thus, if we can establish that languages recognised by groups of the form $A \rtimes \mathbb{Z}_2$, where A is an abelian group, are of generalised star-height at most one then this completes the proof of Theorem 4.1. This result was proved by Pin, Straubing and Thérien [15] in the setting of pseudovarieties. Their proof, which makes use of automata,

Order	Group	Reason
1	$\{1\}$	abelian
2	$\mathbb{Z}_2$	abelian
3	$\mathbb{Z}_3$	abelian
4	$\mathbb{Z}_4$	abelian
	$\mathbb{Z}_2 \times \mathbb{Z}_2$	abelian
5	$\mathbb{Z}_5$	abelian
6	$\mathbb{Z}_6 = \mathbb{Z}_3 \times \mathbb{Z}_2$	abelian
	Dih_3	—
7	$\mathbb{Z}_7$	abelian
8	$\mathbb{Z}_8$	abelian
	$\mathbb{Z}_4 \times \mathbb{Z}_2$	abelian
	$\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$	abelian
	Dih_4	nilpotent of class 2
	Q_8	nilpotent of class 2
9	$\mathbb{Z}_9$	abelian
	$\mathbb{Z}_3 \times \mathbb{Z}_3$	abelian
10	$\mathbb{Z}_{10} = \mathbb{Z}_5 \times \mathbb{Z}_2$	abelian
	Dih_5	—
11	$\mathbb{Z}_{11}$	abelian

Table 4.1: Groups of order less than 12.

is outlined in Section 4.2.1.

At this point, an obvious question to ask is whether languages recognised by finite groups of the form $A \rtimes \mathbb{Z}_r$, where A is an abelian group and r is a positive integer greater than 2, are of generalised star-height at most one. In Section 4.2.2, we explore the case where $r = 3$ by replicating the steps of the proof given for $\mathbb{Z}_2$ in [15]. Unfortunately, this approach does not lead us to the required conclusion but does yield some interesting insights into the problem.

In the upcoming sections, we will make use of star-free injective substitutions and inverse alphabetic homomorphisms, both of which are defined below.

A *substitution* is a function $\sigma : A^* \to \mathcal{P}(B^*)$ such that $\varepsilon\sigma = \{\varepsilon\}$ and $(vw)\sigma = (v\sigma)(w\sigma)$ for all v and w in A^*. The substitution σ is *injective* if $v\sigma \cap w\sigma \neq \emptyset$ implies that $v = w$ for all v and w in A^* and *star-free* if for every star-free language L, the language $L\sigma$ is also star-free.

Proposition 4.2 ([15, Theorem 4.6]). *For every natural number n, the class of languages of generalised star-height at most n is closed under star-free injective* *substitutions.*

A homomorphism $\varphi : A^* \to B^*$ is *alphabetic* if for all a in A, the image of a under φ is either a letter of B or the empty word. For example, the homomorphism $\varphi : \{a,b\}^* \to \{a\}^*$ defined by $a\varphi = a$ and $b\varphi = \varepsilon$ is alphabetic.

Proposition 4.3 ([15, Corollary 4.7])**.** *For every natural number n, the class of languages of generalised star-height at most n is closed under inverse alphabetic morphisms.*

Let $\mathcal{A} = (S, A, s_0, \delta, T)$ be an automaton. Every word $w = a_1 a_2 \dots a_r$ in A^* defines a unique path

$$p(w) = (s_0, a_1)(s_1, a_2) \dots (s_{r-1}, a_r)$$

through $\mathcal{A}$, where we assume that $\delta(s_i, a_{i+1}) = s_{i+1}$ for $0 \le i \le r-1$. Let $|p(w)|_{(s,a)}$ denote the number of times that the arrow (s,a) appears in the path $p(w)$ and define the language $\mathrm{ModCount}(\mathcal{A}, (s,a), k, n)$ by

$$\mathrm{ModCount}(\mathcal{A}, (s,a), k, n) = \{w \in A^* \mid |p(w)|_{(s,a)} \equiv k \pmod{n}\};$$

that is, the set of words w over A such that in the unique path through $\mathcal{A}$ defined by w, the arrow (s,a) is traversed precisely k modulo n times.

An automaton is *transitive* if for all states s_1 and s_2 in S there exists a word w in A^* such that $\delta(s_1, w) = s_2$.

Proposition 4.4 ([15, Proposition 6.3])**.** *Let $\mathcal{A} = (S, A, s_0, \delta, T)$ be a transitive automaton. Then, for every state s in S, every letter a in A, and all integers k and n satisfying $0 \le k < n$, the generalised star-height of the language* $\mathrm{ModCount}(\mathcal{A}, (s,a), k, n)$ *is equal to the generalised star-height of the language* $\mathrm{ModCount}(\mathcal{A}, (s,a), 0, n)$.

Informally, this result states that in transitive automata, counting an arrow k modulo n times is equivalent to counting an arrow 0 modulo n times.

Proposition 4.5 ([15, Proposition 6.4])**.** *Let $\mathcal{A} = (S, A, s_0, \delta, T)$ be a transitive automaton such that for all w in A^*, if $\delta(s,w) = s$ for some state s in S then $\delta(s,w) = s$ for every state s in S. Then, for every state s in S, every letter a in A and every integer $n \ge 2$, the generalised star-height of the language* $\mathrm{ModCount}(\mathcal{A}, (s,a), 0, n)$ *is equal to the generalised star-height of the language* $\mathrm{ModCount}(\mathcal{A}, (s_0,a), 0, n)$.

Informally, this result states that in transitive automata, if, for all words w in A^*, the word w induces the identity on any one state implies that w induces the identity on all states, then counting the arrow (s,a) a multiple of n times is equivalent to counting the arrow (s_0,a) a multiple of n times.

4.2 Semidirect products

Let p be a prime number and let r be a positive integer. Define an automaton $\mathcal{A} = (S, A, s_0, \delta, T)$, where $S = (\mathbb{Z}_p)^r$, the start state s_0 is the r-tuple $(0, 0, \ldots, 0)$, the set of terminal states $T = \emptyset$ and for each a in A there exists an r-tuple t_a in S such that $\delta(s, a) = s + t_a$ for all s in S. Such an automaton is said to be *cyclic*.

Example 4.6. We define a cyclic automaton where $p = r = 2$ and $A = \{a, b, c, d\}$. In this situation, $S = (\mathbb{Z}_2)^2$ and $s_0 = (0, 0)$. Define

$$t_a = t_d = (0,0), \qquad t_b = (0,1) \qquad \text{and} \qquad t_c = (1,0).$$

The transition diagram of our automaton is shown in Figure 4.1.

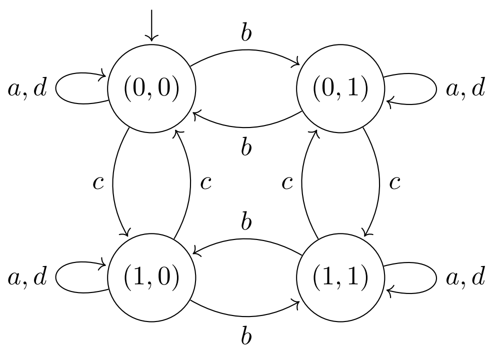

Figure 4.1: Transition diagram for the automaton described in Example 4.6.

Lemma 4.7. *Let $\mathcal{A} = (S, A, s_0, \delta, T)$ be a cyclic automaton, where $S = (\mathbb{Z}_p)^r$. If $\mathcal{A}$ is not transitive then there exists a cyclic automaton $\mathcal{B} = (S, A \cup B, s_0, \delta, T)$ such that the generalised star-height of the language* $\text{ModCount}(\mathcal{A}, (s, a), k, n)$ *is at most the generalised star-height of the language* $\text{ModCount}(\mathcal{B}, (s, a), k, n)$ *for every letter a in A, every state s in S and all integers k and n satisfying $0 \leq k < n$.*

Proof. Suppose that $\mathcal{A}$ is not transitive. Define a new cyclic automaton $\mathcal{B} = (S, A \cup B, s_0, \delta, T)$, where $B = \{b_1, b_2, \ldots, b_r\}$ is a set of letters not in A and

$$\delta((z_1, z_2, \ldots, z_r), b_i) = (z_1, z_2, \ldots, z_{i-1}, z_i + 1, z_{i+1}, z_{i+2}, \ldots, z_r)$$

for $1 \leq i \leq r$. The automaton $\mathcal{B}$ is transitive as each b_i induces a cycle through p different states and each state appears in at least two distinct cycles. Moreover,

$$\text{ModCount}(\mathcal{A}, (s, a), k, n) = (\text{ModCount}(\mathcal{B}, (s, a), k, n) \cap A^*)\varphi^{-1},$$

where $\varphi : A^* \to (A \cup B)^*$ is the natural alphabetic homomorphism defined by $w\varphi = w$ for all w in A^*. Now, A^* is a star-free subset of $(A \cup B)^*$ by Lemma 1.24 and, since we are closed under inverse alphabetic homomorphisms by Proposition 4.3, the result follows. □

Lemma 4.7 tells us that we need only consider transitive cyclic automata from this point onwards. Thus, in all of our upcoming results, we can assume that $k = 0$ and $s = s_0$ by Propositions 4.4 and 4.5.

The following result, for which the proof is omitted, shows that counting a loop in a cyclic automaton results in a language of generalised star-height at most one.

Proposition 4.8 ([15, Proposition 6.8])**.** *Let $\mathcal{A} = (S, A, s_0, \delta, T)$ be a cyclic automaton and let a in A induce the identity on S. For all integers k and $n \geq 2$ with $0 \leq k < n$ and for every state s in S, the generalised star-height of the language* $\mathrm{ModCount}(\mathcal{A}, (s, a), k, n)$ *is at most one.*

4.2.1 Cyclic group of order 2

In one specific situation, we can improve the result of Proposition 4.8.

Proposition 4.9 ([15, Proposition 6.10])**.** *Let $\mathcal{A} = (S, A, s_0, \delta, T)$ be a cyclic automaton, where $S = (\mathbb{Z}_2)^r$. For all integers k and $n \geq 2$ with $0 \leq k < n$, every letter a in A and every state s in S, the generalised star-height of the language* $\mathrm{ModCount}(\mathcal{A}, (s, a), k, n)$ *is at most one.*

We dedicate the rest of this section to the proof of Proposition 4.9 and a second result that will complete the proof of Theorem 4.1.

By an argument similar to that in the omitted proof of Proposition 4.8, we can assume that $r = 1$, so that $S = \{0, 1\}$. Fix a letter a in A and partition A into three subsets as follows: $\{a\}$; C, the set of all letters inducing the identity on S; and, $B = A \backslash (\{a\} \cup C)$. The transition diagram of $\mathcal{A}$ is shown in Figure 4.2.

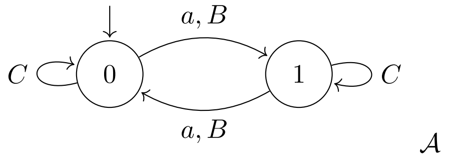

Figure 4.2: Transition diagram for the automaton $\mathcal{A}$ in Proposition 4.9.

Let $\varphi : A^* \to \{a,b\}^*$ be the alphabetic homomorphism defined by

$$a\varphi = a \qquad \text{and} \qquad x\varphi = \begin{cases} b, & \text{if } x \in B, \\ \varepsilon, & \text{if } x \in C, \end{cases}$$

where x is any element of $A \setminus \{a\}$. Let $\mathcal{A}' = (S, \{a,b\}, 0, \gamma, T)$ be the cyclic automaton defined by the transitions

$$\gamma(0,a) = 1, \qquad \gamma(1,a) = 0, \qquad \gamma(0,b) = 1 \qquad \text{and} \qquad \gamma(1,b) = 0.$$

The transition diagram of $\mathcal{A}'$ is shown in Figure 4.3.

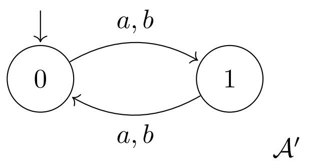

Figure 4.3: Transition diagram for the automaton $\mathcal{A}'$ in Proposition 4.9.

Let $p(w)$ (respectively, $p'(w)$) be the path defined by a word w in $\mathcal{A}$ (respectively, $\mathcal{A}'$). Then, for every w in A^*,

$$|p(w)|_{(0,a)} = |p'(w\varphi)|_{(0,a)}.$$

Hence,

$$\mathrm{ModCount}(\mathcal{A}, (0,a), k, n) = (\mathrm{ModCount}(\mathcal{A}', (0,a), k, n))\varphi^{-1}.$$

Since, by Proposition 4.3, the class of languages of generalised star-height at most n is closed under inverse alphabetic morphisms, it suffices to show that the generalised star-height of the language $\mathrm{ModCount}(\mathcal{A}', (0,a), k, n)$ is at most one.

Let $L = \mathrm{ModCount}(\mathcal{A}', (0,a), k, n)$ and

$$X = \{w \in \{a,b\}^* \mid \gamma(0,w) = 0\},$$

and consider the set $K = L \cap X$; that is, the set of words w in L such that the path in $\mathcal{A}'$ traversed by w begins and ends at state 0. We claim that we can write L as a right quotient of K; specifically,

$$L = K\{\varepsilon, b\}^{-1}.$$

To see this, let w be a word in L and let $\gamma(0,w) = s$. If $s = 0$ then w lies in $K = K\varepsilon^{-1}$. Otherwise, $s = 1$ and $\gamma(0,wb) = \gamma(1,b) = 0$ and $|p'(wb)|_{(0,a)} = |p'(w)|_{(0,a)}$. Hence, wb lies in K and w lies in Kb^{-1}.

Conversely, let w be a word from $K\{\varepsilon, b\}^{-1}$. Then, either w lies in K or wb lies in K. In the first case, it follows from the definition of K that w lies in L. In the second case, $\gamma(0, wb) = 0$ which implies that $\gamma(0, w) = 1$. It follows that $|p'(wb)|_{(0,a)} = |p'(w)|_{(0,a)}$ and, therefore, w lies in L, proving the claim.

Since, by Proposition 1.22, the class of languages of generalised star-height at most n is closed under right quotients, it suffices to show that the generalised star-height of K is at most one.

From the transition diagram of $\mathcal{A}'$, we see that if we start at state 0 and read either an a or a b then we end up in state 1. From state 1, we have no choice but to return to state 0 by reading either an a or a b. Thus,

$$X = ((a \cup b)^2)^* = (aa \cup ab \cup ba \cup bb)^*.$$

Using this, we find that a naïve expression for K is given by

$$((ba \cup bb)^*(aa \cup ab))^k(((ba \cup bb)^*(aa \cup ab))^n)^*(ba \cup bb)^*. \tag{4.1}$$

In this form, the expression has generalised star-height two. However, if we could write $(ba \cup bb)^*$ as a star-free expression then the generalised star-height of the expression given in (4.1) would reduce to one. Unfortunately, $(ba \cup bb)^*$ does not have an equivalent star-free expression. This can be established by finding the minimal automaton for $(ba \cup bb)^*$, calculating its transition monoid, noting that it is finite and aperiodic and appealing to Schützenberger's Theorem (Theorem 1.23).

However, we can think about K in a different way. Indeed, we notice that every word in X can be uniquely factorised into consecutive subwords of length two. For example, the word

$$w = aababbab = aa \cdot ba \cdot bb \cdot ab. \tag{4.2}$$

With slight abuse of notation, for w in X, we denote by $|w|_x$ the number of occurrences of x in the unique factorisation of w as a product of words of length two. For example, in Equation (4.2), we see that

$$|w|_{aa} = |w|_{ab} = |w|_{ba} = |w|_{bb} = 1.$$

It follows that

$$\begin{aligned} K &= \{w \in X \mid |w|_{aa} + |w|_{ab} \equiv k \pmod n\} \\ &= \{w \in X \mid 2|w|_{aa} + 2|w|_{ab} \equiv 2k \pmod{2n}\}. \end{aligned}$$

At this point, we notice that

$$2|w|_{aa} + |w|_{ab} + |w|_{ba} = |w|_a$$

for every word w in X and, therefore, we can write K as

$$K = \{w \in X \mid |w|_a + |w|_{ab} - |w|_{ba} \equiv 2k \pmod{2n}\}$$

by replacing the $2|w|_{aa}$ term with its equivalent expression.

We can now decompose K into a boolean combination of languages of the form

$$\{w \in X^* \mid |w|_a \equiv m \pmod{2n}\}$$

and languages of the form

$$\{w \in X^* \mid |w|_{ab} - |w|_{ba} \equiv m \pmod{2n}\}.$$

Languages of the first type are recognised by a finite abelian group by Theorem 2.2 and are of generalised star-height at most one. Though not shown here, languages of the second type are dealt with via a star-free injective substitution and are of generalised star-height at most one. Taking a boolean combination of these languages does not increase the generalised star-height and, therefore, K is of generalised star-height at most one. Thus, $\mathrm{ModCount}(\mathcal{A}, (s, a), k, n)$ is of generalised star-height at most one, which concludes the proof of Proposition 4.9.

Proposition 4.9 is the main component in the proof of the following theorem:

Theorem 4.10 ([15, Theorem 7.6])**.** *Every language recognised by a monoid of the pseudovariety* $\mathbf{AbGrps} \rtimes \mathbb{Z}_2$ *is of generalised star-height at most one.*

It follows from Theorem 4.10 that languages recognised by $\mathrm{Dih}_3 = \mathbb{Z}_3 \rtimes \mathbb{Z}_2$ and $\mathrm{Dih}_5 = \mathbb{Z}_5 \rtimes \mathbb{Z}_2$ are all of generalised star-height at most one. This completes the proof of Theorem 4.1.

4.2.2 Cyclic group of order 3

When trying to determine the generalised star-height of languages recognised by a finite group of a given order, Theorem 4.1 implies that the first hurdle we need to overcome is those languages recognised by groups of order 12. In Table 4.2, we list the groups of orders 12, 13, 14 and 15, noting that all but two of these groups have already been resolved.

The two remaining groups, namely A_4 and Dic_3, can be decomposed as semidirect products of an abelian group with a cyclic group. In the first case, the cyclic group is $\mathbb{Z}_3$ while in the second case the cyclic group is $\mathbb{Z}_4$. At this point, we make the following conjecture.

Conjecture 4.11. *Every language recognised by a monoid of the pseudovariety* $\mathbf{AbGrps} \rtimes \mathbb{Z}_3$ *is of generalised star-height at most one.*

Order	Group	Reason
12	$\mathbb{Z}_{12} = \mathbb{Z}_4 \times \mathbb{Z}_3$	abelian
	$\mathbb{Z}_6 \times \mathbb{Z}_2$	abelian
	$\mathrm{Dih}_6 = \mathbb{Z}_6 \rtimes \mathbb{Z}_2$	Theorem 4.10
	$A_4 = (\mathbb{Z}_2 \times \mathbb{Z}_2) \rtimes \mathbb{Z}_3$	—
	$\mathrm{Dic}_3 = \mathbb{Z}_3 \rtimes \mathbb{Z}_4$	—
13	$\mathbb{Z}_{13}$	abelian
14	$\mathbb{Z}_{14} = \mathbb{Z}_7 \times \mathbb{Z}_2$	abelian
	$\mathrm{Dih}_7 = \mathbb{Z}_7 \rtimes \mathbb{Z}_2$	Theorem 4.10
15	$\mathbb{Z}_{15} = \mathbb{Z}_5 \times \mathbb{Z}_3$	abelian

Table 4.2: Groups of order 12, 13, 14 and 15.

In order to prove this conjecture, it suffices to prove the following result.

Conjecture 4.12. *Let $\mathcal{A} = (S, A, s_0, \delta, T)$ be a cyclic automaton, where $S = (\mathbb{Z}_3)^r$. For all integers k and $n \geq 2$ with $0 \leq k < n$, every letter a in A and every state s in S, the generalised star-height of the language* $\mathrm{ModCount}(\mathcal{A}, (s, a), k, n)$ *is at most one.*

If Conjecture 4.12 is true then we can prove Conjecture 4.11 in exactly the same way that Proposition 4.9 proves Theorem 4.10. Unfortunately, we cannot prove Conjecture 4.12 using the same proof ideas that were used in Proposition 4.9. We use the remainder of this section to explain why this method of proof fails when we replace the state set $S = (\mathbb{Z}_2)^r$ with $(\mathbb{Z}_3)^r$.

By a similar argument to that in the omitted proof of Proposition 4.8, we can assume that $r = 1$, so that $S = \{0, 1, 2\}$. Fix a letter a in A and partition A into four subsets as follows: $\{a\}$; D, the set of all letters inducing the identity on S; C, the set of letters inducing the permutation $(0\ 2\ 1)$ on S; and, $B = A \setminus (\{a\} \cup C \cup D)$. The transition diagram of $\mathcal{A}$ is shown in Figure 4.4.
Let $\varphi : A^* \to \{a, b, c\}^*$ be the alphabetic homomorphism defined by

$$a\varphi = a \qquad \text{and} \qquad x\varphi = \begin{cases} b, & \text{if } x \in B, \\ c, & \text{if } x \in C, \\ \varepsilon & \text{if } x \in D, \end{cases}$$

where x is any element of $A \setminus \{a\}$. Let $\mathcal{A}' = (S, \{a, b, c\}, 0, \gamma, T)$ be the cyclic

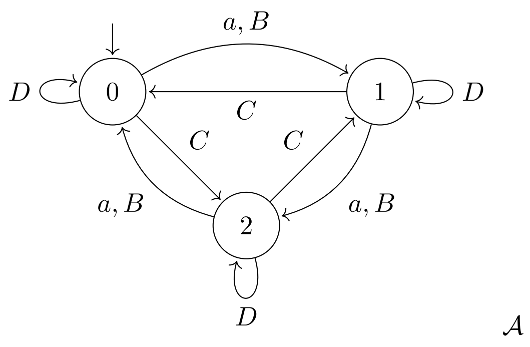

Figure 4.4: Transition diagram for the automaton $\mathcal{A}$ in Conjecture 4.12.

automaton defined by the transitions

$$\begin{aligned} &\gamma(0,a) = 1, \qquad \gamma(1,a) = 2, \qquad \gamma(2,a) = 0, \\ &\gamma(0,b) = 1, \qquad \gamma(1,b) = 2, \qquad \gamma(2,b) = 0, \\ &\gamma(0,c) = 2, \qquad \gamma(1,c) = 0, \qquad \text{and} \qquad \gamma(2,c) = 1. \end{aligned}$$

The transition diagram of $\mathcal{A}'$ is shown in Figure 4.5.

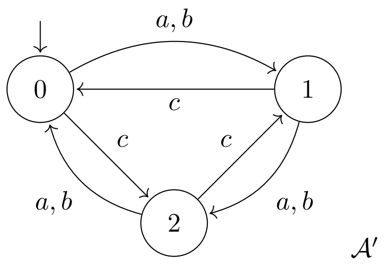

Figure 4.5: Transition diagram for the automaton $\mathcal{A}'$ in Conjecture 4.12.

Let $p(w)$ (respectively, $p'(w)$) be the path defined by a word w in $\mathcal{A}$ (respectively, $\mathcal{A}'$). Then, for every w in A^*,

$$|p(w)|_{(0,a)} = |p'(w\varphi)|_{(0,a)}.$$

Hence,

$$\text{ModCount}(\mathcal{A}, (0,a), k, n) = (\text{ModCount}(\mathcal{A}', (0,a), k, n))\varphi^{-1}.$$

Since, by Proposition 4.3, the class of languages of generalised star-height at most n is closed under inverse alphabetic morphisms, it suffices to show that the generalised star-height of the language $\text{ModCount}(\mathcal{A}', (0,a), k, n)$ is at most one.

Let $L = \mathrm{ModCount}(\mathcal{A}', (0,a), k, n)$ and

$$X = \{w \in \{a,b,c\}^* \mid \gamma(0,w) = 0\},$$

and consider the set $K = L \cap X$; that is, the set of words w in L such that the path in $\mathcal{A}'$ traversed by w begins and ends at state 0. We claim that we can write L as a right quotient of K; specifically,

$$L = K\{\varepsilon, b, c\}^{-1}.$$

To see this, let w be a word in L and let $\gamma(0,w) = s$. If $s = 0$ then w lies in $K = K\varepsilon^{-1}$. If $s = 1$ then $\gamma(0,wc) = \gamma(1,c) = 0$ and $|p'(wc)|_{(0,a)} = |p'(w)|_{(0,a)}$. Hence, wc lies in K and w lies in Kc^{-1}. Otherwise, $s = 2$ and $\gamma(0,wb) = \gamma(2,b) = 0$ and $|p'(wb)|_{(0,a)} = |p'(w)|_{(0,a)}$. Hence, wb lies in K and w lies in Kb^{-1}.

Conversely, let w be a word from $K\{\varepsilon, b, c\}^{-1}$. Then, either w lies in K, wb lies in K or wc lies in K. In the first case, it follows from the definition of K that w lies in L. In the second case, $\gamma(0,wb) = 0$ which implies that $\gamma(0,w) = 2$. It follows that $|p'(w)|_{(0,a)} = |p'(wb)|_{(0,a)}$ and, therefore, w lies in L. Finally, if wc lies in K then $\gamma(0,wc) = 0$, which implies that $\gamma(0,w) = 1$. It follows that $|p'(w)|_{(0,a)} = |p'(wc)|_{(0,a)}$ and, therefore, w lies in L. This completes the proof of the claim.

Since, by Proposition 1.22, the class of languages of generalised star-height at most n is closed under right quotients, it suffices to show that the generalised star-height of K is at most one.

From the transition diagram of $\mathcal{A}'$, we see that if we start at state 0 and read an a or a b then we travel to state 1. From here, we can alternate between states 1 and 2 by reading either an a or a b followed by a c. This always brings us back to state 1. In order to return to state 0 from state 1, we can either read a c or read an a or a b to travel to state 2 before reading either an a or a b to travel to state 0. This can be summed up by the following expressions:

$$E = a((a \cup b)c)^*((a \cup b)^2 \cup c) \qquad \text{and} \qquad F = b((a \cup b)c)^*((a \cup b)^2 \cup c).$$

Alternatively, we can start at state 0 and read the letter c. This moves us to state 2. From here, we can alternate between states 2 and 1 by reading a c followed by either an a or a b. This always brings us back to state 2. In order to return to state 0 from state 2 we can either read an a or a b or read a c to travel to state 1 before reading another c to travel to state 0. This can be summed up by the following expression:

$$G = c(c(a \cup b))^*(a \cup b \cup c^2).$$

It follows that $X = (E \cup F \cup G)^*$. Using this, we find that a naïve expression for K is given by

$$((F \cup G)^* E)^k (((F \cup G)^* E)^n)^* (F \cup G)^*. \tag{4.3}$$

In this form, the expression has generalised star-height three, since E, F and G all contain one star. However, $((a \cup b)c)^*$ and $(c(a \cup b))^*$ both have equivalent star-free expressions by Schützenberger's Theorem (Theorem 1.23) and, therefore, E, F and G are languages of generalised star-height zero. This means that the expression given in (4.3) is actually of generalised star-height two. Unfortunately, $(F \cup G)^*$ does not have an equivalent star-free expression (established, again, through the use of Schützenberger's Theorem) and therefore we cannot reduce the generalised star-height of the expression given in (4.3) any further.

It is at this point that following the strategy employed in the proof of Proposition 4.9 fails. In this situtation, we cannot uniquely factorise the words in X into consecutive subwords of a fixed length and cannot, therefore, write K as a boolean combination of subword counting languages. This is because each of the individual components in the union that make up X are infinite languages as opposed to finite languages, as previously seen. As such, Conjecture 4.12 remains open.

Attempting to employ the same proof strategy when the state set S is $(\mathbb{Z}_4)^r$ would be futile as reducing the problem to $r = 1$ relies on the primality of the modulus p. Thus, determining the generalised star-height of languages recognised by groups of the form $A \rtimes \mathbb{Z}_3$ and $A \rtimes \mathbb{Z}_4$ remains an open problem.

Chapter 5

Conclusions and Open Questions

In this final chapter, we present a number of open questions that have been raised by the work in this thesis alongside questions raised by the existing results concerning the Generalised Star-Height Problem.

In Chapter 2, we saw Thérien's Theorem (Theorem 2.1) which establishes a link between languages recognised by finite nilpotent groups of a given class and languages of the form $\mathrm{ScatModCount}(w, k, n)$. By Eilienberg's Variety Theorem (Theorem 1.17), we know that there exists a one-to-one correspondence between the collection of all pseudovarieties of monoids and the collection of all varieties of monoid languages. Even though we established in Section 2.7 that languages of the form $\mathrm{ModCount}(w, k, n)$ do not form a variety of monoid languages, they might generate one. Thus, it is natural to ask the following question:

Question 5.1. *Is a variety of monoid languages generated by languages of the form* $\mathrm{ModCount}(w, k, n)$ *and, if so, what is it? In other words, is there a corresponding Thérien-like theorem for contiguous subwords; that is, is there a corresponding statement of Theorem 2.1 for contiguous subwords?*

Our original motivation for counting subwords stemmed from the scattered subword ordering that can be placed on the set of all words over a given alphabet. Most known results concerning subword counting relate to this ordering, as we saw in Section 2.1. Theorem 2.5 shows that establishing the generalised star-height of the languages recognised by nilpotent groups of class three is incomplete. Moreover, Theorem 2.4 is the only known result that contributes towards establishing the generalised star-height of the languages recognised by nilpotent groups of class four or higher. Thus, we are interested in finding answers to the following questions:

Question 5.2. *What is the generalised star-height of a language of the form* $\mathrm{ScatModCount}(abc, k, n)$*, where* a*,* b *and* c *are letters from an alphabet and* n *is not a square-free integer.*

Question 5.3. *What is the generalised star-height of a language of the form* $\mathrm{ScatModCount}(w, k, n)$*, where* $|w| \geq 4$*?*

Answering these two questions would allow us to establish the generalised star-height of any language recognised by a finite nilpotent group due to the connection found in Thérien's Theorem (Theorem 2.1).

The scattered subword ordering and the contiguous subword ordering are just two examples of orders that can be placed on the set of all words over a given alphabet. An example of another ordering that we can place on the set of all words over a given alphabet is the *embedding order* [1, Definition 3.1], which is defined as follows: v is an *embedded subword* of w if and only if w can be created from v by inserting new occurrences of letters into v after they have first appeared in v. For example, ab is a embedded subword of aba but is not an embedded subword of bab. For every subword ordering we can define notions of $\mathrm{Count}(w, k)$ and $\mathrm{ModCount}(w, k, n)$, which leads to the following question:

Question 5.4. *If we equip the set of all words over an alphabet with a different subword ordering then what is the generalised star-height of the corresponding* $\mathrm{Count}(w, k)$ *and* $\mathrm{ModCount}(w, k, n)$ *languages?*

In Chapter 3, we began a classification of the languages that are recognised by finite semigroups. In order to do this, we appealed to The Rees Theorem (Theorem 3.1) and studied languages recognised by Rees zero-matrix semigroups over certain finite groups. In particular, we established that the generalised star-height of the languages recognised by Rees zero-matrix semigroups over finite nilpotent groups of classes zero and one is at most one. Thus, it is natural to ask the following question:

Question 5.5. *Which languages are recognised by Rees zero-matrix semigroups over finite nilpotent groups of class two? What is the generalised star-height of each of these languages?*

Given the amount of work required to answer the equivalent question when class two is replaced by class one, it would not be amiss to expect this question to have a difficult and lengthy proof. As such, we may wish to restrict ourselves to a specific example of a nilpotent group of class two in the first instance. Hence, we pose the following question:

Question 5.6. *Which languages are recognised by Rees zero-matrix semigroups over the dihedral group* Dih_4*? What is the generalised star-height of each of these languages?*

To finish Chapter 3, we determined the generalised star-height of the languages recognised by Rees zero-matrix semigroups over monogenic semigroups. Using the results of Section 3.4.3, we can extend this result to direct products of monogenic semigroups. Now, monogenic semigroups are a subclass of commutative semigroups but, unfortunately, unlike in the case of groups, direct products of finite monogenic semigroups do not exhaust all finite commutative semigroups. We raise the following question:

Question 5.7. *Which languages are recognised by Rees zero-matrix semigroups over commutative semigroups? What is the generalised star-height of each of these languages?*

In Chapter 4, we considered which languages are recognised by finite groups of a given order. This work was motivated by Theorem 4.1 which states that every language recognised by a finite group of order less than 12 is of generalised star-height at most one. The proof of this result relies on semidirect products of the form $A \rtimes \mathbb{Z}_2$, where A is an aperiodic monoid, and on cyclic automata. When trying to extend this to semidirect products of the form $A \rtimes \mathbb{Z}_3$, the proof strategy breaks down. This means that the following question remains open:

Question 5.8. *Which languages are recognised by a monoid from the pseudovariety* $\mathbf{AbGrps} \rtimes \mathbb{Z}_r$*, where* r *is an integer greater than or equal to* 3*, and what is the generalised star-height of each of these languages?*

The first major step towards classifying the languages recognised by finite groups is to complete the case for groups of order 12. From existing results, three of the five groups have been dealt with and the generalised star-height of the languages that they recognise has been established. It remains to answer the following question:

Question 5.9. *Which languages are recognised by the finite groups* A_4 *and* Dic_3*? What is the generalised star-height of each of these languages?*

Providing an answer to this question would then allow us to state the generalised star-height of every language recognised by a group of order up to and including 15.

Perhaps the most pertinent question in need of an answer is that of the Generalised Star-Height Problem itself; that is,

Question 5.10 (Generalised Star-Height Problem). *Does there exist an algorithm to determine the generalised star-height of a regular language. In particular, does there exist a language of generalised star-height greater than one?*

When we consider all of the known results and all of the languages introduced in this thesis, we see that the generalised star-height is at most one in every case. As such, the Generalised Star-Height Problem still awaits a solution.

Bibliography

[1] E. Aichinger, P. Mayr, and R. McKenzie. On the number of finite algebraic structures. *Journal of the European Mathematical Society*, 16:1673–1686, 2014.

[2] T. Bourne and N. Ruškuc. On the star-height of subword counting languages and their relationship to Rees zero-matrix semigroups. *Theoretical Computer Science*, 653:87–96, 2016.

[3] F. Dejean and M. P. Schützenberger. On a question of Eggan. *Information and Control*, 9:23–25, 1966.

[4] L. C. Eggan. Transition graphs and the star-height of regular events. *Michigan Mathematical Journal*, 10:385–397, 1963.

[5] S. Eilenberg. *Automata, Lanaguages, and Machines.* Academic Press, Vol.B, 1976.

[6] K. Hashiguchi. Representation theorems on regular languages. *J. Comput. System Sci.*, 27:101–115, 1983.

[7] W. H. Henneman. *Algebraic theory of automata.* PhD thesis, MIT, 1971.

[8] J. M. Howie. *Automata and Languages.* Oxford University Press, 1991.

[9] J. M. Howie. *Fundamentals of Semigroup Theory.* Oxford University Press, 1995.

[10] T. W. Hungerford. *Algebra.* Springer-Verlag, 1974.

[11] D. Kirsten. Distance desert automata and the star height problem. *RAIRO - Theoretical Informatics and Applications*, 39:455–509, 2005.

[12] S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata studies*, pages 3–42. Princeton University Press, 1956.

[13] M. V. Lawson. *Finite Automata.* Chapman & Hall/CRC, 2004.

[14] J. E. Pin. Syntactic semigroups. In G. Rozenberg and A. Salomaa, editors, *Handbook on Formal Languages Vol.1: Word, Language, Grammar*, chapter 10, pages 679–746. Springer-Verlag, 1997.

[15] J. E. Pin, H. Straubing, and D. Thérien. Some results on the generalized star-height problem. *Information and Computation*, 101:291–250, 1992.

[16] J. Sakarovitch. *Elements of Automata Theory.* Cambridge University Press, 2009.

[17] M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.

[18] D. Thérien. Subword counting and nilpotent groups. In L. J. Cummings, editor, *Combinatorics on Words: Progress and Perspectives*, pages 297–305. Academic Press, 1983.