CLASSIFICATION AND ENUMERATION OF FINITE SEMIGROUPS

Andreas Distler

A Thesis Submitted for the Degree of PhD at the University of St. Andrews

2010

Ph.D. Thesis

University of St Andrews

Classification and Enumeration of Finite Semigroups

Andreas Distler

St Andrews, May 2010

Declarations

I, Andreas Distler, hereby certify that this thesis, which is approximately 40,000 words in length, has been written by me, that it is the record of work carried out by me and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student and as a candidate for the degree of Doctor of Philosophy in February 2006; the higher study for which this is a record was carried out in the University of St Andrews between 2006 and 2010.

Date: Signature of candidate:

I, Nik Ruškuc, hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of Doctor of Philosophy 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:

In submitting this thesis to the University of St Andrews we understand that we are giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. We also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. We have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the electronic publication of this thesis:

Access to Printed copy and electronic publication of thesis through the University of St Andrews.

Date: Signature of candidate:

Date: Signature of supervisor:

Abstract

The classification of finite semigroups is difficult even for small orders because of their large number. Most finite semigroups are nilpotent of nilpotency rank 3. Formulae for their number up to isomorphism, and up to isomorphism and anti-isomorphism of any order are the main results in the theoretical part of this thesis. Further studies concern the classification of nilpotent semigroups by rank, leading to a full classification for large ranks.

In the computational part, a method to find and enumerate multiplication tables of semigroups and subclasses is presented. The approach combines the advantages of computer algebra and constraint satisfaction, to allow for an efficient and fast search. The problem of avoiding isomorphic and anti-isomorphic semigroups is dealt with by supporting standard methods from constraint satisfaction with structural knowledge about the semigroups under consideration. The approach is adapted to various problems, and realised using the computer algebra system GAP and the constraint solver Minion. New results include the numbers of semigroups of order 9, and of monoids and bands of order 10. Up to isomorphism and anti-isomorphism there are 52,989,400,714,478 semigroups with 9 elements, 52,991,253,973,742 monoids with 10 elements, and 7,033,090 bands with 10 elements. That constraint satisfaction can also be utilised for the analysis of algebraic objects is demonstrated by determining the automorphism groups of all semigroups with 9 elements.

A classification of the semigroups of orders 1 to 8 is made available as a data library in form of the GAP package Smallsemi. Beyond the semigroups themselves a large amount of precomputed properties is contained in the library. The package as well as the code used to obtain the enumeration results are available on the attached DVD.

Contents

List of Tables

List of Figures

Preface

Classification is an important research tool in all sciences. When concerned with different objects or a certain phenomenon, which comes in different shapes, one wants to bring order into the chaos by determining common features and differences. Classification does so by building a catalogue of types of objects or phenomena. Many such catalogues have been successfully utilised in science and mathematics. An example not too far from the research area in this thesis demonstrates how beneficial such a catalogue can be: the classification of finite simple groups is used in the proofs of numerous results in group theory.

Algebraic structures often appear in different representations. This is fine as long as one studies each object by itself, but causes difficulties if objects are to be compared. The most important decision in a classification is which criteria are used to distinguish objects. The representation is usually not a criterion for the classification, and in particular one wants to be able to determine when two objects in different representations have the same structure. What this actually means depends not only on the type of object, but as well on the interest one has in it. A circle, for example, has topologically the same structure as a square, though any non-mathematician would point out clear differences between the two.

The objects studied in this thesis are finite semigroups. The work started with the attempt to use a classification of semigroups of small order to analyse their automorphism groups, only to find out that such a classification was not readily available (see Section 4.2). The sheer number of semigroups makes a complete classification difficult, and makes it even more difficult to provide one in a useful format. The number of semigroups grows super-exponentially with their order (see [KRS76]). Where the number of semigroups makes a classification impossible, the related task of counting the semigroups might be achievable. Enumeration is less informative than classification, but can provide more than just a number: for different subclasses of semigroups numbers can be compared; or, if the enumeration is done with a formula, probabilities and asymptotics may be established.

The major existing results in the classification and enumeration of semigroups are summarised in the forthcoming section. The subsequent section provides details on the content of this thesis.

History

The classification of finite semigroups was first done by hand. Tamura classified the semigroups up to order 4 [Tam53, Tam54] (published in 1953 and 1954), and classified, together with Tetsuya, Hashimoto, Akazawa, Shibata, and Inui, the semigroups up to order 5 [THA+55] in the year after. The latter contained at least one mistake in the original version. This became apparent from a comparison with results from a computer search for semigroups. The first such was done by Forsythe in 1954 to obtain the semigroups of order 4 [For55]. One year later the semigroups of order 5 were found by Motzkin and Selfridge [MS55, For60]. From that point on the semigroups of increasing order were too numerous to be considered by hand computations, and even presenting more than their number became difficult. Plemmons created multiplication tables for the semigroups of order 6 in 1966 [Ple67, Ple70], and stored them on magnetic tape, so that they could be analysed. Much later, in 1989, Jürgensen wrote a report annotating these tables [Jür89]. In between Jürgensen and Wick counted the semigroups of order 7 [JW77], but did not store the multiplication tables. The same is likely to be true for Satoh, Yama, and Tokizawa who counted the semigroups of order 8 and partitioned them according to their structure [SYT94]. This last result was published in 1994. The most recent publication on an attempt to improve the enumeration methods is from 2007 [Gri07], but it does not contain new numbers.

Naturally, many of the authors mentioned in the previous paragraph commented on the challenge to enumerate the next higher order. Their guesses tend to be rather far away from the actual numbers. This has changed since Kleitman, Rothschild, and Spencer determined a lower bound for the number of semigroups on a finite set, which they attempt to prove asymptotic [KRS76]. Even though details of the proof are omitted the fact is widely believed.[1] The result also led to

[1] This opinion is shared by various semigroup theorists the author has talked to. A written

a lower bound for the number of structural types of semigroups [JMS91, Chapter 15], which appears to be tight. The majority of semigroups for orders 7 and 8 are of the type used in [KRS76], and their ratio in semigroups of a given order seems to converge to 1 while the order tends to infinity.

This completes the summary of major results which are extended in this thesis. Closely related research has been done for the special case of commutative semigroups. The most recent results are available in [Gri03]. Two older surveys on the use of computers in semigroup theory, including enumeration of semigroups, are [Ple69, Jür78].

Content

The thesis contains theoretical and computational results on the classification and enumeration of finite semigroups and subclasses like monoids, bands, and nilpotent semigroups.

After some introductory material is presented in the next chapter, the first new results are given in Chapter 2. The main theorems provide formulae for the number of 3-nilpotent semigroups. Kleitman, Rothschild, and Spencer state in [KRS76] that the ratio of the number of such semigroups on the set $\{1, 2, \ldots, n\}$ and the number of all semigroups on $\{1, 2, \ldots, n\}$ tends to 1 while $n \in \mathbb{N}$ tends to infinity. The conjecture, that the analogue is true for the numbers of structural types of semigroups, is supported by empirical evidence in [SYT94]. The formulae from Chapter 2 play an important role for the enumeration of semigroups in later chapters.

The thesis continues with a chapter which serves on the one hand as preparation for the computer search of semigroups presented later, and contains on the other hand some theoretical results connected to this preparation. In the first part of Chapter 3 an algorithm to construct diagonals of multiplication tables is given, based on a correspondence between diagonals and certain digraphs. The diagonal of the multiplication table defines the squares of elements. The second section of Chapter 3 gives a partial answer to the question which structural information of a semigroup can be deduced from the squares of all elements.

Chapters 4 and 5 are dedicated to the computer search for semigroups. The

account is given in [JMS91].

method used for the search differs from what had been done before. Instead of a specialised implementation, general purpose software is facilitated. The idea is to formulate the problem of finding all semigroups as a constraint satisfaction problem (CSP). Constraint satisfaction is an area in computer science which provides a framework to solve all kinds of (real life) problems, which allow a combinatorial description. The CSP – and hence the original problem of finding semigroups – is then solved with the constraint solver Minion [GJM06]. This approach separates the search into a black box process, so that one can concentrate on the setup of the CSP. Not having to worry too much about the search, makes it possible to put all effort in a sophisticated setup. This becomes particularly important if one considers subclasses of semigroups, for which additional structural information is available. Already a very simple setup allows one to reproduce on a modern computer the known results from the enumeration of semigroups up to order 8 in Chapter 4. More elaborate formulations using mathematical knowledge about semigroups are developed in Chapter 5. The computer algebra system GAP [GAP08] is used for algebraic manipulations required for the setup. In addition, a technique, translating and extending an idea used by Plemmons in the enumeration of semigroups of order 6 [Ple67], which splits one CSP into a family of CSPs solving the same problem, is introduced and applied.

While the first part of Chapter 4 describes how the semigroups up to order 8 were created, the second part reports on the construction of a data library in GAP containing their multiplication tables and information on some of their properties. The data library is available as an add-on for GAP, the package Smallsemi [DM10], which is contained on the DVD attached to this thesis.

Chapter 5 contains six sections. In the first section the split of a CSP into a family of CSPs is explained. In the three subsequent sections the method is applied to bands, semigroups, and monoids respectively, leading to new results in the enumeration of all three types of algebraic objects. Finally, it was possible to go back to the starting point of this thesis and compute the automorphism groups of all semigroups with at most 9 elements. Possible future applications of the developed methods are discussed in the closing section.

A verification of the enumerative results is possible using the code included on the attached DVD. One idea behind the presented approach is that anybody can reproduce the results using their chosen CSP solver. As stated earlier, Minion was used as a black box to solve carefully formulated CSPs, and so could (potentially) any other CSP solver. This is not the whole truth though. As Minion is developed at the University of St Andrews, where this thesis was written, there has been an active interaction between the author and the developers of Minion. Over the course of the PhD project Minion became orders of magnitude more efficient. This was partially due to feedback from the intensive computations undertaken for this thesis. While, for example, the results on the number of semigroups of order up to 8 are reproduced in Chapter 4 using the simplest formulation of the problem, this would not have been possible with the version of Minion available three years ago. On the other hand, using the most sophisticated setup, the results in Chapter 5 can be obtained with an old version of Minion on a four years old desktop computer. Unfortunately, the improvements to Minion and the benefits from mathematical knowledge put in the setup are often not cumulative.

Acknowledgements

Remembering what brought me to St Andrews in first place, I am grateful that Prof. Joachim Neubüser contacted Prof. Edmund Robertson, and that the latter invited me and helped me to overcome my initial doubts.

I want to thank my first supervisor Prof. Nik Ruškuc for all the guidance and support he gave me, and my second supervisor Prof. Steve Linton for all his advice. My main collaborators Dr Tom Kelsey and Dr James Mitchell I thank for the fruitful work together and for all their help.

I had the great pleasure to share 'my' office with so many people. They are Dr Robert Brignall, Anne-Sandrine Paumier, Markus Pfeiffer, Claire Pollard, Victor Maltcev, and Dr Stephen Waton. Thanks go to all of them, but in particular to Victor, who is the office mate, from whose knowledge I benefited most.

Of all the other current and former members of CIRCA, who I owe thanks for their help, I want to mention Dr John McDermott, who dedicated numerous hours to deal with computer issues occurring during my rather extensive computations and my work as GAP packages administrator.

I also thank my parents, Dagmar and Dietrich Distler, and my sister Denise for their short and long distance support over the last four years.

Thanks go to my friends in St Andrews for reminding me that there is a life outside university; and to my more distant friends for reminding me that there is a world outside St Andrews.

I am especially grateful for the invaluable support and encouragement I received from Ana Patricia Barazal Barreira – who deserves a title far more than I do.

Last but not least I acknowledge the financial support I received from the University of St Andrews and from CIRCA, sponsoring my PhD and the attendance of many conferences.

St Andrews, Andreas Distler
February 2010

With all the changes for the final version of my thesis done, I want to thank my internal examiner Dr Max Neunhöffer and my external examiner Prof. Rick Thomas for reading the thesis and for the corrections they suggested. Even more though, I am grateful for the sympathy they showed and for the effort they made which allowed me to complete my studies this month.

St Andrews, Andreas Distler
May 2010

1 Mathematical Background

This chapter provides background information for the rest of the thesis. Basic definitions and well-known facts are presented. Conventions for the notation used throughout the thesis are introduced. The chapter divides into four sections. The areas covered are binary operations in general, semigroups in particular, group actions, and directed graphs.

None of the sections is intended as an introductory text to the respective area, but to enable a mathematician to understand this work without having to consult the literature. Recommendations on further reading are given throughout.

1.1 Binary Operations

The algebraic objects encountered in this thesis are sets with a binary operation defined on them, most often semigroups. It is assumed that the reader is familiar with the basic definitions of algebraic structures. As a reminder and to introduce the notation which will be used, the current section starts with a summary of the basic definitions. For more detailed information the reader is referred to [How95, Chapter 1] or [Kos82, Chapter 4], or other introductory books on algebra or semigroup theory.

Definition 1.1.1 Let X be a set.

(i) A mapping $* : X \times X \to X$ is a *(binary) operation*, and the pair $(X, *)$ is a *magma*.

(ii) A magma $(X, *)$ and the operation $*$ are *associative*, if $(a * b) * c = a * (b * c)$ for all $a, b, c \in X$. An associative magma is a *semigroup*.

(iii) An element $e \in X$ is an *identity (element)* of a magma $(X, *)$, if $e * x = x$ and $x * e = x$ for all $x \in X$. A semigroup containing an identity is a *monoid*.

(iv) An element $x \in X$ in a magma $(X, *)$ is *invertible*, if there exists an identity $e \in X$ and an element $\bar{x} \in X$ for which $x * \bar{x} = e$ and $\bar{x} * x = e$. Then $\bar{x}$ is an *inverse* of x. A monoid in which every element is invertible is a *group*.

The cardinality of X is called the *order* or *size* of the magma (semigroup, monoid, and group respectively).

It is common practice to identify the magma $(X, *)$ with the underlying set X and to denote the operation by simple juxtaposition of elements. Only in cases where this is potentially ambiguous, operation symbols will be used. In general the operation will be referred to as multiplication. If a magma contains an identity, then it is unique. If an element x in a monoid is invertible, then it has a unique inverse, which will be denoted x^{-1}.

The way in which algebraic objects are represented is explained next. The order of any algebraic object studied in this thesis is finite and most commonly denoted by $n \in \mathbb{N}$ (where $\mathbb{N} = \{1, 2, \dots\}$). The most naïve way to describe a binary operation on a finite set is to note the result of the operation for every pair of elements. This is usually done in the form of a table, the *multiplication table* or – particularly for groups – *Cayley table* of a magma. The table is a square matrix of dimension n whose rows and columns are indexed by the elements in the set. Each entry defines the product of its row index multiplied with its column index. That is, if $(X, *)$ is a magma, then its multiplication table $T_{(X,*)} = (t_{a,b})_{a,b \in X}$ has $a * b$ as entry $t_{a,b}$ for all $a, b \in X$. Figure 1.1 shows as example the Cayley table of the group $(\{e, c\}, *)$ of order 2, where e is the identity element.

$*$	e	c
e	e	c
c	c	e

Figure 1.1 Cayley table of a group with two elements

Most often the underlying set of a magma will be $\{1, 2, \dots, n\}$, which shall be abbreviated as $[n]$. As a consequence one can assume that the rows and columns of the multiplication table are indexed according to their position in the table. This allows one to omit the row and column header. Under this convention every square matrix of size n with entries in $[n]$ uniquely defines a binary operation on $[n]$. The set of all such matrices will be denoted by Ω_n. Figure 1.2 shows the Cayley table of V_4 (the Klein four-group) on $\{1, 2, 3, 4\}$ with 1 as identity element.

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{pmatrix}$$

Figure 1.2 A Cayley table of V_4

Even for finite algebraic objects it is in general not a good idea to represent them via their multiplication tables. The representation is not particularly instructive to learn about the structure of the object, and the data to be retained at once is in general unreasonably large. For magmas it is feasible if they are of such small orders as predominately occurring in this thesis. Then the simplicity of the representation makes it attractive for use with computers. The product of any two elements is immediately available and membership is not an issue, while storing tables of this size is not a problem.

When studying algebraic objects one is usually more interested in their structure than in their specific representation. This is reflected by an equivalence relation defined on algebraic objects of every type, placing them in the same equivalence class, if they have the same structure. This shall be made precise for magmas.

Definition 1.1.2 Let $(X, *)$ and $(Y, \circ)$ be two magmas of the same order. A bijection $\sigma : X \to Y$ is an *isomorphism* if it respects the multiplication – that is, $\sigma(a * b) = \sigma(a) \circ \sigma(b)$ for all $a, b \in X$ – and an *anti-isomorphism* if it reverses the multiplication – that is, $\sigma(a * b) = \sigma(b) \circ \sigma(a)$ for all $a, b \in X$. If such a bijection exists, then the magmas are *isomorphic*, respectively *anti-isomorphic*, and are *equivalent* if they are either isomorphic or anti-isomorphic.

Loosely speaking, an isomorphism does nothing, but rename the elements in the underlying set, and an anti-isomorphism changes in addition multiplication from the left with multiplication from the right. Neither is essential for the structural properties of the magma. Hence enumeration is done up to equivalence in this thesis. To provide more complete information, results from enumeration up to equivalence, obtained in the main body of the thesis, are given up to isomorphism

$$\begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 4 & 1 & 2 \\ 2 & 1 & 4 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix} \xrightarrow{\text{apply}(1\,4)} \begin{array}{c|cccc} & 4 & 2 & 3 & 1 \\ \hline 4 & 1 & 3 & 2 & 4 \\ 2 & 3 & 1 & 4 & 2 \\ 3 & 2 & 4 & 1 & 3 \\ 1 & 4 & 2 & 3 & 1 \end{array} \xrightarrow{\text{rearrange}} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{pmatrix}$$

Figure 1.3

A Cayley table of V_4, with 4 as identity, is mapped to a Cayley table of V_4, where 1 is the identity, using the permutation $(1\,4)$.

in Appendix A.3. For a set $\mathcal{X}$ of magmas, $|\overline{\mathcal{X}}|$ shall denote the number of non-equivalent magmas, and $|\widehat{\mathcal{X}}|$ the number of non-isomorphic magmas in $\mathcal{X}$.

Instead of asking whether two magmas are isomorphic, one can determine an isomorphic magma under a specific bijection. Given a magma $(X, *)$, a set Y and a bijection $\sigma : X \to Y$, define a multiplication $\circ$ on Y as follows: for $s, t \in Y$ with $s = \sigma(a)$ and $t = \sigma(b)$ the product $s \circ t$ equals $\sigma(a * b)$. This makes σ an isomorphism from $(X, *)$ to $(Y, \circ)$. If $T_{(X,*)} = (t_{a,b})_{a,b \in X}$ is the multiplication table of $(X, *)$, then

$$(\sigma\left(a * b\right))_{\sigma(a),\sigma(b) \in Y} = \left(\sigma\left(\sigma^{-1}(s) * \sigma^{-1}(t)\right)\right)_{s,t \in Y} \tag{1.1}$$

is the multiplication table of $(Y, \circ)$. What happens to the table is that every element is mapped to its image under the bijection, and this includes the indices. To comply with the convention that each index equals the position of the row respectively column in the table, rows and columns have to be rearranged. This is best illustrated in an example as given in Figure 1.3, which splits the process into two steps. First the table is mapped to its image disregarding the convention, and then the rows and columns are put in the correct order.

The magma $(Y, \cdot)$ which – in the same setup as before – is anti-isomorphic to $(X, *)$, has

$$\left(\sigma\left(\sigma^{-1}(t) * \sigma^{-1}(s)\right)\right)_{s,t \in Y} \tag{1.2}$$

as multiplication table. If a bijection is applied as an anti-isomorphism to a multiplication table, then the matrix is transposed in addition to the steps undertaken for the isomorphism corresponding to the same bijection.

The *dual* of a magma X, denoted by $X^\perp$, is the image of X under the anti-isomorphism corresponding to the trivial permutation on X. A magma which is anti-isomorphic to itself – or equivalently, isomorphic to its dual – is *self-dual*. Note that in a group every anti-isomorphism gives rise to an isomorphism by combining it with taking inverses. Combining the identity mapping with taking inverses shows that every group is anti-isomorphic to itself (since $(ab)^{-1} = b^{-1}a^{-1}$) and hence self-dual. If in a classification of magmas up to equivalence the self-dual ones are known, it allows one to deduce the corresponding classification up to isomorphism.

Lemma 1.1.3 *Let $\mathcal{X}$ be a set of non-equivalent magmas. A set of non-isomorphic magmas in $\mathcal{X} \cup \{X^\perp \mid X \in \mathcal{X}\}$ is given by $\mathcal{X} \cup \{X^\perp \mid X \in \mathcal{X}, X$ is not self-dual$\}$.*

Proof: Since the magmas in $\mathcal{X}$ are non-equivalent, the possible pairs of isomorphic magmas in $\mathcal{X} \cup \{X^\perp \mid X \in \mathcal{X}\}$ are a magma and its dual. Such magmas are by definition self-dual. $\square$

Note that for self-dual objects it does not make a difference whether they are considered up to isomorphism or up to equivalence. This holds in particular for commutative objects.

1.2 Semigroups

In this section terminology and well-known results around properties of semigroups and their structure are presented. Again the reader is directed to other sources, in particular [How95] and [CP61], for a general introduction to semigroup theory.

In the previous section a binary operation was given by its multiplication table. Another common way of representing an algebraic object is to find an embedding into an existing object. According to Cayley's theorem every finite group of order n is isomorphic to a permutation group on n elements, thus can be embedded into the full symmetric group on $[n]$. An analogue statement holds for semigroups.

Lemma 1.2.1 *Let S be a semigroup of order $n \in \mathbb{N}$. Then S is isomorphic to a subsemigroup of the full transformation monoid on $[n+1]$ (all mappings from $[n+1]$ into itself).*

A proof for a more general statement, including the case that S is infinite, can be found in [How95, Theorem 1.1.2].

A semigroup S is *generated by* $A = \{a_1, a_2, \ldots, a_k\}$ if $A \subseteq S$ and all elements in S are products of finite length of elements from A. The elements in A are *generators*, A is a *generating set*, and one writes $S = \langle A \rangle$ or $S = \langle a_1, a_2, \ldots, a_k \rangle$. When presenting a semigroup S as a subsemigroup of a full transformation monoid it makes sense to give just the generators of S, as the multiplication is determined by the embedding. A set A, for which $\langle A \setminus \{a\} \rangle$ is a proper subset of $\langle A \rangle$ for all $a \in A$ is a *minimal generating set* of $\langle A \rangle$.

If a semigroup S is generated by a single element, then S is *monogenic*. If $S = \langle a \rangle = \{a^i \mid i \in \mathbb{N}\}$ is a finite semigroup, then there exist minimal $m, r \in \mathbb{N}$ such that $a^m = a^{m+r}$. The integers m and r are *index* and *period* of a and of S, and $S = \{a, a^2, \ldots, a^{m+r-1}\}$. Hence

$$m + r = |S| + 1. \tag{1.3}$$

For each pair of positive integers, $m, r \in \mathbb{N}$, there exists a monogenic semigroup with index m and period r. The transformation

$$\begin{pmatrix} 1 & 2 & 3 & \cdots & m+r+1 \\ r & 1 & 2 & \cdots & m+r \end{pmatrix} \tag{1.4}$$

generates such a semigroup. Every element in a semigroup generates a monogenic subsemigroup.

A subsemigroup I of a semigroup S is an *ideal* if every product containing an element from I as factor lies in I.

The order of a finite semigroup has essentially no relevance for its structure – very much on the contrary to the situation for groups. The following lemma, copied from [JMS91], gives some insight into this comment.

Lemma 1.2.2 *Let S, T be two non-isomorphic semigroups of order $n \in \mathbb{N}$. Then there exist for all $i \in \mathbb{N} \cup \{0\}$ two non-isomorphic semigroups S_i and T_i of order $n+i$, such that S_i contains S as subsemigroup and T_i contains T as subsemigroup.*

To prove the lemma, define based on a semigroup S a monoid S^1 with an additional element, which is an identity.

Proof: The statement is shown by induction. Let $S_0 = S$ and $T_0 = T$. By assumption, S_0 and T_0 have the required properties. Now consider $i > 0$ and assume S_{i-1} and T_{i-1} fulfil the condition from the lemma. Let $S_i = S_{i-1}^1$ and $T_i = T_{i-1}^1$.

Assume $\sigma : S_i \to T_i$ is an isomorphism. Since σ must map the identity in S_i to the identity in T_i an isomorphism from S_{i-1} to T_{i-1} is induced by restricting σ to S_{i-1}. The existence of such an isomorphism contradicts the induction hypothesis. Hence S_i and T_i are non-isomorphic. □

Structural statements about semigroups are therefore usually independent of their size.

Remark 1.2.3 In every finite semigroup there exists at least one element which is equal to its own square, and hence equal to all its powers [How95, Proposition 1.2.3]. Such an element is called an *idempotent*. A semigroup consisting entirely of idempotents is a *band*.

The set of idempotents in a magma X is denoted $E(X)$. One particular type of idempotent, an identity, has already appeared in Definition 1.1.1. Let X be a magma and $e \in E(X)$. If $ex = x$ ($xe = x$) for all $x \in X$, then e is a *left (right) identity*. If $ex = e$ ($xe = e$) for all $x \in X$, then e is a *left (right) zero*. An idempotent, which is both a left and a right zero is a *zero (element)* of the magma. While identities and zeros are unique, left/right identities or zeros may not be. A magma in which every element is a left (right) zero element is a *left (right) zero semigroup* – which is indeed a semigroup – and every element is a right (left) identity, too. A magma with zero element, in which every product equals the zero, is a *zero semigroup* – and is again indeed a semigroup.

Further properties of semigroups, which are solely mentioned, but not used, in the course of this thesis, are compiled in Appendix B.

For finite semigroups the most important tool to describe their structure are four equivalence relations on the elements, known as Green's relations. Two elements a, b in a semigroup S are *$\mathcal{L}$-related*, denoted by $a\mathcal{L}b$, if there exist $s, t \in S^1$ such that $sa = b$ and $tb = a$. This defines an equivalence $\mathcal{L}$ on S. Analogously one defines an equivalence $\mathcal{R}$. Two elements a, b in a semigroup S are *$\mathcal{R}$-related*, if there exist $s, t \in S^1$ such that $as = b$ and $bt = a$. Furthermore, two elements are

$\mathcal{H}$*-related*, if they are both $\mathcal{L}$- and $\mathcal{R}$-related. Finally, if there exists $c \in S$ such that $a\mathcal{L}c$ and $c\mathcal{R}b$, then a and b are $\mathcal{D}$*-related*. The classes defined by these four Green's relations are $\mathcal{H}$-, $\mathcal{L}$-, $\mathcal{R}$-, and $\mathcal{D}$*-classes* respectively. All four equivalence relations are preserved under isomorphism.

Under an anti-isomorphism the Green's relations $\mathcal{H}$ and $\mathcal{D}$ are preserved, while $\mathcal{L}$ and $\mathcal{R}$ are interchanged. The latter relations are only invariant as an unordered tuple. This is one reason why in certain situations one might be interested in semigroups up to isomorphism instead of up to equivalence.[1] The connecting link are the self-dual semigroups as shown in Lemma 1.1.3, since they are anti-isomorphic to itself.

1.3 Group Actions

The effect isomorphisms and anti-isomorphisms have on a multiplication table is a special case of the common group theoretical notion of an action. Plenty of detailed introductions are available in the literature, for example [Kos82, Chapter 7, §2].

Definition 1.3.1 Let X be a set and G a group with identity e. A mapping

$$\phi : X \times G \to X, (x, g) \mapsto x^g$$

is a *right action* if $x^e = x$ and $x^{gh} = (x^g)^h$ for all $x \in X$ and for all $g, h \in G$. A *left action* is defined analogously.

The *orbit* of an element $x \in X$ is the set $\{x^g \mid g \in G\}$, denoted x^G. Similarly for $Y \subseteq X$ define $Y^g = \{x^g \mid x \in Y\}$ and $Y^G = \bigcup_{x\in Y} x^G$.

The *stabiliser* of an element $x \in X$ is the set $\{g \in G \mid x^g = x\}$, denoted by $\mathrm{Stab}_G(x)$. The *pointwise stabiliser* of a subset $Y \subseteq X$ is the intersection of the stabilisers of the elements in Y, that is $\bigcap_{x\in Y} \mathrm{Stab}_G(x)$. The *setwise stabiliser* of Y, denoted by $\mathrm{Stab}_G(Y)$, is the set $\{g \in G \mid Y^g = Y\}$.

Each action of a group G on a set induces an equivalence relation. The equivalence classes are the orbits of the elements in the set. Elements in the same orbit are called G*-equivalent*.

[1] It is not a coincidence that only $\mathcal{H}$- and $\mathcal{D}$-relations play a role in the search for semigroups up to equivalence in Chapter 5.

Equivalence of magmas arising from multiplication tables in Ω_n (the set of all $n \times n$ matrices with entries in $[n]$) can now be expressed using the language from the previous definition. For a set X denote by S_X the *(full) symmetric group* of all permutations on X and denote $S_n = S_{[n]}$. Consider the action $\Omega_n \times S_n \to \Omega_n$ sending the table $T = (t_{i,j})_{1 \le i,j \le n} \in \Omega_n$ under $\pi \in S_n$ to $(t^{\pi}_{i^{\pi^{-1}},j^{\pi^{-1}}})_{1 \le i,j \le n}$. Then two tables $T_1, T_2 \in \Omega_n$ define isomorphic magmas if and only if there exists a bijection $\pi \in S_n$ sending T_1 to T_2. This follows from Equation (1.1) and the definition of the action.

So far the action only covers isomorphisms. Anti-isomorphisms correspond to the permutations in S_n as well. To have the action cover both isomorphisms and anti-isomorphisms at the same time a simple trick is used. Let $S_n \times C_2$ act on the set Ω_n. Group elements (π, e), where e is the identity in C_2, act like π in the previously described action, thus correspond to isomorphisms. For (π, c) instead, where c is the non-trivial element in C_2, the action on a table $(t_{i,j})_{1 \le i,j \le n} \in \Omega_n$ is $(t^{\pi}_{j^{\pi^{-1}},i^{\pi^{-1}}})_{1 \le i,j \le n}$, following (1.2). This reflects the fact that an anti-isomorphism reverses the order of the multiplication. Remember that the multiplication table is transposed in addition to what π does as an isomorphism. A simple example is given in Figure 1.4.

$$\begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}$$

Figure 1.4 Action of an anti-isomorphism on a multiplication table
The multiplication table of a right-zero semigroup with two elements goes to the multiplication table of a left-zero semigroup under every anti-isomorphism.

Using the defined actions two magmas are in the same orbit, if they have the same structure. Here 'the same structure' means that the magmas are isomorphic, if they are S_n-equivalent, and it means that the magmas are isomorphic or anti-isomorphic, if they are $S_n \times C_2$-equivalent. Note that the partition of Ω_n into orbits induced by S_n is a refinement of the partition induced by $S_n \times C_2$. Moreover, the same considerations hold for every subset of Ω_n that is closed under the action, in particular it holds for the set of matrices defining semigroups. The two multiplication tables in Figure 1.4 form one orbit under the action of $S_2 \times C_2$, and are each an orbit by itself under the action of S_2.

Definition 1.3.2 Let $\phi : X \times G \to X$ be an action, and let $\psi : X \to Y$ be a function. If the mapping $\phi^\psi : \mathrm{im}(\psi) \times G \to \mathrm{im}(\psi)$ with $(y, g) \mapsto \psi(x^g)$ for $y = \psi(x)$ is well-defined, then ϕ^ψ is an *induced action*.

Induced actions from the action of $S_n \times C_2$ on multiplication tables will play an important role in this thesis.

1.4 Directed Graphs

The elements of graph theory which appear in this thesis are, together with the notation, mostly taken from [HP73]. The reference is concerned with enumeration of graphs, and adapting the techniques presented in it will turn out to be useful for the enumeration of semigroups in this thesis (see Sections 2.2 and 2.3).

The graphs occurring in this thesis have directed, but no multiple edges, and loops are allowed. Some definitions are given next to introduce the notation. A *digraph* is a pair (V, E) formed by a set V of *vertices* and a set $E \subseteq V \times V$ of edges. If $e = (v, w) \in E$, then v is the *start vertex* of the edge e and w is the *end vertex*. The *outdegree* of a vertex $v \in V$ is the number of edges with v as start vertex, and the *indegree* is the number of edges with v as end vertex.

Let $v_0, v_1, \ldots, v_k$ be a sequence of vertices such that (v_{i-1}, v_1) is an edge for all $1 \leq i \leq k$. Then the sequence together with the edges is a *walk of length* k. If all the vertices are distinct, then the walk is a *path*. A walk in which the first and the last vertex coincide, and all other vertices are pairwise distinct is a *cycle*.

Let $\Gamma = (V, E)$ be a graph and $U \subseteq V$. Then Γ *without* U, denoted $\Gamma \setminus U$, is the graph $(V \setminus U, E')$, where $E' = \{e \in E \mid e = (v, w) \text{ with } v, w \in V \setminus U\}$. If $\Delta = (W, F)$ is a second graph, then the *union* $\Gamma \cup \Delta$ is the graph $(V \cup W, E \cup F)$.

A bijection $\sigma : V \to W$ between the vertex sets of two graphs, $\Gamma = (V, E)$ and $\Delta = (W, F)$, is an *isomorphism (of graphs)* if: $(u, v) \in E$ if and only if $(\sigma(u), \sigma(v)) \in F$. If such a bijection exist, then Γ and Δ are *isomorphic*.

Graphs for which the direction of edges does not matter only appear as underlying structures. The *underlying graph* of a digraph $\Gamma = (V, E)$ is $(V, E \cup \bar{E})$, where $\bar{E} = \{(w, v) \mid (v, w) \in E\}$. A *connected component* of a digraph $\Gamma = (V, E)$ is a subset U of V such that there exists a path between any two vertices of U in the underlying graph, and for each edge $(v, w) \in E$ either $v, w \in U$ or $v, w \in V \setminus U$. A digraph is *connected*, if its vertex set is a connected component.

2 Nilpotent Semigroups

In a constructive approach to the enumeration of algebraic objects, based on mathematical knowledge about their structure, it is necessary to understand certain characteristics of the objects. Part of this is to look for properties which the majority of the objects have in common. In this chapter finite semigroups S for which the sequence of sets $S \supseteq S^2 \supseteq S^3 \supseteq \ldots$ (where $S^k = \{s_1 s_2 \cdots s_k \mid s_i \in S\}$) stabilises with a singleton set are studied.

Definition 2.0.1 A semigroup S is *nilpotent* if there exists an $r \in \mathbb{N}$ such that $|S^r| = 1$. The least such r is the *nilpotency rank* of S and S is said to be r*-nilpotent*.

As observed in [SYT94], 99.5% of the semigroups of order 8 are nilpotent and 99.4% even 3-nilpotent. In this chapter a classification of all finite, nilpotent semigroups is attempted. For 3-nilpotent semigroups the studies lead to the most important theoretical result of this thesis: a formula for the number of 3-nilpotent semigroups up to isomorphism and anti-isomorphism. Additional results are obtained for nilpotent semigroups with nilpotency rank at least $n-2$, where n denotes the order of the semigroups. A complete classification is not achieved using the developed methods – even when restricting to small orders, which are of primary interest for the enumeration in later chapters.

The chapter is organised in three sections. It starts with an introduction to nilpotent semigroups and results for nilpotency ranks not equal to 3. The major part of the chapter is concerned with the enumeration of 3-nilpotent semigroups. The techniques used are compiled in Section 2.2, while the results form the final section. Formulae for 3-nilpotent semigroups are established up to isomorphism and up to equivalence, and for the subclass of commutative semigroups.

2.1 Nilpotency Rank

This section contains results related to the classification of nilpotent semigroups by nilpotency rank.

For finite semigroups there is an equivalent formulation of the definition of nilpotency, involving only powers of elements instead of all possible products.

Lemma 2.1.1 *Let S be a finite semigroup. Then the following are equivalent:*

(i) *S is nilpotent;*

(ii) *S contains a zero element z and for every element $s \in S$ there exists $k_s \in \mathbb{N}$ such that $s^{k_s} = z$.*

Proof: (i) $\Rightarrow$ (ii) : If S is r-nilpotent and z denotes the unique element in S^r, then $sz = zs = z$ for all $s \in S$ as $zs, sz \in S^{r+1} = S^r$. Therefore z is a zero element. Moreover, $s^r = z$ for all $s \in S$, showing that the second statement follows from the first.

(ii) $\Rightarrow$ (i) : Let $k = \max\{k_s \mid s \in S\}$ and $n = |S|$. Consider the product $s_1 s_2 \cdots s_{n+1}$ of length $n+1$ with each factor in S. Then the sequence of prefixes, $(s_1 \cdots s_i)_{1 \leq i \leq n+1}$, must contain a repetition, say $s_1 \cdots s_l = s_1 \cdots s_m$ with $l < m \leq n+1$. Repeatedly replacing $s_1 \cdots s_l$ with $s_1 \cdots s_m$ leads to

$$s_1 \cdots s_l = s_1 \cdots s_l(s_{l+1} \cdots s_m) = \cdots = s_1 \cdots s_l(s_{l+1} \cdots s_m)^k = s_1 \cdots s_l z = z.$$

As l is strictly smaller than m and hence at most n, consequently $s_1 s_2 \cdots s_n = z$. Thus all products of length at least n equal z, showing that S is n-nilpotent. $\square$

For the forward direction of the previous lemma finiteness of the semigroup is not required. Note that in general $k = \max\{k_s \mid s \in S\}$ from the proof does not equal the nilpotency rank of S. The proof shows that n is an upper bound for the nilpotency rank of a semigroup S with n elements. Another way to see this, is to consider the sequence of sets $S \supseteq S^2 \supseteq S^3 \supseteq \ldots$ mentioned in the introduction of this chapter. The sequence becomes constant with the first repetition of a set, which therefore occurs with S^{r+1} in the case of an r-nilpotent semigroup. As the sequence starts with an n element set this yields $r \leq n$.

Lemma 2.1.2 *Let S be an r-nilpotent semigroup. Then the following hold:*

(i) *the sets* $S^k \setminus S^{k+1}$ *with* $1 \leq k \leq r-1$ *are non-empty and form a partition of* $S \setminus S^r$*;*

(ii) *if* $s = s_1 s_2 \cdots s_k \in S^k \setminus S^{k+1}$ *with* $1 \leq k \leq r-1$*, then* $s_i \cdots s_j \in S^{j-i+1} \setminus S^{j-i+2}$ *for all* $1 \leq i \leq j \leq k$*.*

Proof: (i): For any three sets A, B, C with $A \supseteq B \supseteq C$ the set $A \setminus C$ equals the disjoint union of $A \setminus B$ and $B \setminus C$. Hence it suffices to show that the sets $S^k \setminus S^{k+1}$ are non-empty for $1 \leq k \leq r-1$. As explained in the comments before the lemma there is no repetition in $S \supseteq S^2 \supseteq \ldots \supseteq S^r$. Thus S^{k+1} is a proper subset of S^k for $1 \leq k \leq r-1$.

(ii): The statement is shown by contradiction. Assume that $s_i \cdots s_j \in S^{j-i+2}$ for some $1 \leq i \leq j \leq k$. This means $s_i \cdots s_j$ can be expressed as a product $t_1 \cdots t_{j-i+2} \in S^{j-i+2}$. Replacing $s_i \cdots s_j$ by $t_1 \cdots t_{j-i+2}$ in s, that is

$$s = s_1 s_2 \cdots s_k = s_1 \cdots s_{i-1} t_1 \cdots t_{j-i+2} s_{j+1} \cdots s_k,$$

yields $s \in S^{k+1}$, a contradiction. □

Both parts of the previous lemma are used repeatedly in the proofs throughout this section. It is worthwhile to gain some intuition for its content. First of all, for each element other than the zero, the length of a product equalling the element is restricted. An element in $S^k \setminus S^{k+1}$ can be written as product with k factors, but not as product with $k+1$ factors. Collecting in separate sets elements with the same maximal length of a product equalling the element yields a partition of a nilpotent semigroup. An r-nilpotent semigroup is partitioned into r sets: one for each maximal length between 1 and $r-1$; and the zero element in a set by itself. Looking at it this way each part of a product of maximal length is clearly maximal itself, which is essentially what is stated in the second part of the lemma. This shows in particular that every element in a product of maximal length is in $S \setminus S^2$. This connects to a further well-known result.

Corollary 2.1.3 *Let* S *be a semigroup. Then the set* $S \setminus S^2$ *is contained in any minimal generating set. If* S *is nilpotent and has size at least* 2*, then* $S \setminus S^2$ *is the unique minimal generating set.*

Proof: The statement for the general case is obvious. It remains to be shown that $S \setminus S^2$ is a generating set if S with $|S| \geq 2$ is nilpotent. The zero element of such a semigroup is not contained in any minimal generating set as every element has a power that equals the zero. According to Lemma 2.1.2(i) each element $s \in S \setminus S^r$ is in $S^k \setminus S^{k+1}$ for some $1 \leq k \leq r-1$ and thus has an expression as a product of k factors. By choosing $i = j$ in Lemma 2.1.2(ii) it follows that all the factors are in $S \setminus S^2$. □

Further consequences of Lemma 2.1.2 are restrictions on the sizes of the sets $S^k \setminus S^{k+1}$ partitioning a nilpotent semigroup S.

Corollary 2.1.4 *Let S be an r-nilpotent semigroup of order n. Then the following hold:*

(i) *if* $|S^l \setminus S^{l+1}| = 1$ *for any* $1 \leq l \leq r-1$ *then* $|S^k \setminus S^{k+1}| = 1$ *for all* $l \leq k \leq r-1$;

(ii) $|S^k \setminus S^{k+1}| \leq \min_{1\leq i\leq k-1}\{|S^i \setminus S^{i+1}|\,|S^{k-i} \setminus S^{k-i+1}|\}$ *for all* $2 \leq k \leq r-1$.

Proof: (i): Let $s = s_1 s_2 \cdots s_k \in S^k \setminus S^{k+1}$ for some $l < k \leq r-1$. According to Lemma 2.1.2(ii) the product $s_1 s_2 \cdots s_l$ of length l equals the unique element $t_1 t_2 \cdots t_l$ in $S^l \setminus S^{l+1}$. Hence $s_1 s_2 \cdots s_k = t_1 t_2 \cdots t_l s_{l+1} \cdots s_k$. The right hand side of this equation is still a product of maximal length equalling s. Thus $t_2 t_3 \cdots t_l s_{l+1}$ equals as well the unique element in $S^l \setminus S^{l+1}$ and can be replaced by $t_1 t_2 \cdots t_l$. Applying this argument repeatedly and always replacing the product of length l with $t_1 t_2 \cdots t_l$ yields $s = t_1^{k-l+1} t_2 \cdots t_l$. This implies $|S^k \setminus S^{k+1}| = 1$, since $s \in S^k \setminus S^{k+1}$ was chosen arbitrarily.

(ii): Let $s = s_1 s_2 \cdots s_k \in S^k \setminus S^{k+1}$ for some $2 \leq k \leq r-1$. For any $1 \leq i \leq k-1$ it follows with Lemma 2.1.2(ii) that $s_1 \cdots s_i \in S^i \setminus S^{i+1}$ and $s_{i+1} \cdots s_k \in S^{k-i} \setminus S^{k-i+1}$. Hence, $|S^i \setminus S^{i+1}|\,|S^{k-i} \setminus S^{k-i+1}|$ is an upper bound for the number of possible values of s. □

Applying the structural information provided by Lemma 2.1.2 and its corollaries to nilpotent semigroups of certain rank leads to the first results in the classification of finite nilpotent semigroups.

Lemma 2.1.5 *Let S be a semigroup of order n. Then the following are equivalent:*

(i) *S is n-nilpotent;*

(ii) *S is monogenic and nilpotent;*

(iii) *S is monogenic with index n and period 1.*

Proof: (i) $\Rightarrow$ (ii): According to Lemma 2.1.2(i) the sets $S^k \setminus S^{k+1}$ with $1 \leq k \leq n$ are non-empty. Hence each set contains exactly one element. The set $S \setminus S^2$ is a generating set due to Corollary 2.1.3 and thus S is monogenic.

(ii) $\Rightarrow$ (iii): If a denotes the generator of S, then a^n and a^{n+1} both equal the zero element. Therefore the period of S is 1. Solving Equation (1.3) shows the index equals n.

(iii) $\Rightarrow$ (i): If a denotes the generator of S, then the equality $a^n = a^{n+1}$ holds. Consequently a^n is a zero and every element to some power equals a^n. According to Lemma 2.1.1 S is nilpotent. Moreover, $S^k \setminus S^{k+1} = \{a^k\}$ for $1 \leq k \leq n-1$, showing that S has nilpotency rank n. $\square$

Lemma 2.1.5 makes it possible to classify nilpotent semigroups with at most 3 elements. The trivial semigroup is 1-nilpotent by Definition 2.0.1 and in agreement with the previous result. Every semigroup with more than one element has nilpotency rank at least 2. Moreover, in a 2-nilpotent semigroup every product equals the zero element. Hence for order 2 the zero semigroup is the only nilpotent semigroup, and for order 3 there is one 2-nilpotent and one 3-nilpotent semigroup.

Remark 2.1.6 For any order greater than 1 the zero semigroup is the unique 2-nilpotent semigroup of that order.

To classify nilpotent semigroups with n elements and nilpotency rank $n-1$ for larger n, conclude from Lemma 2.1.5 that for each nilpotent semigroup of order n with rank less than n, the minimal generating set contains at least two elements; and exactly two if the rank equals $n-1$ due to Lemma 2.1.2(i). The latter leads to the following result.

Theorem 2.1.7 *Let S be a nilpotent semigroup of order n ($n \neq 4$) and nilpotency rank $n-1$. Then $S = T \cup \{x\}$ where T is a monogenic subsemigroup of S with nilpotency rank $n-1$.*

Proof: The statement is trivially true for $1 \leq n \leq 3$. For $n \geq 5$ let $S = \langle u, v \rangle$ (see the comment before the theorem), and let t_k denote the unique element in $S^k \setminus S^{k+1}$ for $2 \leq k \leq n-2$, where uniqueness follows from Lemma 2.1.2(i). At least one of uv, vu, u^2, and v^2 equals t_2. Assume neither u^2 nor v^2 does. Hence, following Lemma 2.1.2(ii), uu and vv do not appear in any product of maximal length 3 equalling t_3. Remember in addition that all factors in a product of maximal length are generators, then of all elements in S^3 only uvu or vuv might equal t_3. Again it follows from Lemma 2.1.2(ii) that $t_2 = uv = vu$ in both cases. Thus $t_3 = u(vu) = u(uv) = u^2v$ or $t_3 = v(uv) = v(vu) = v^2u$, a contradiction to the assumption that neither u^2 nor v^2 equal t_2.

Without loss of generality let $t_2 = u^2$. If $t_k = s_1 s_2 \cdots s_k$ for any $2 \leq k \leq n-2$, then $s_i s_{i+1} = t_2$ for all $1 \leq i \leq k-1$, once more due to Lemma 2.1.2(ii). Hence $t_k = u^k$ for $2 \leq k \leq n-2$. Finally u^{n-1} equals the zero element as S has nilpotency rank $n-1$. Hence $T = \langle u \rangle$ is a nilpotent, monogenic subsemigroup with $n-1$ elements. Then T has nilpotency rank $n-1$ according to Lemma 2.1.5. By choosing $x = v$ the statement follows. $\square$

For semigroups of order 4 the step in the proof of the previous theorem that restricts the possible results for products of two elements to certain combinations does not work. Indeed, every combination can occur as will be shown in Section 2.3 studying 3-nilpotent semigroups in general.

The statement about the structure of the type of semigroups in Theorem 2.1.7 makes it possible to classify, and hence to count, those semigroups.

Theorem 2.1.8 *For $n \geq 5$ there are $n + \lfloor n/2 \rfloor$ nilpotent semigroups with n elements and nilpotency rank $n-1$.*

Proof: Let $S = \langle u, v \rangle$ and $t_k = u^k \in S^k \setminus S^{k+1}$ for $2 \leq k \leq n-2$ be as in the proof of the previous theorem. If in addition the values of uv, vu, and v^2 are known, S is uniquely determined, because $u^k v$ and vu^k can then be deduced for all $2 \leq k \leq n-1$. Since S is nilpotent, possible values for uv, vu, and v^2 are elements in $S^2 = \{u^k \mid 2 \leq k \leq n-1\}$. The choices not contradicting associativity are determined below by considering several cases. Moreover, choices leading to equivalent semigroups are identified. Note that any isomorphism or anti-isomorphism preserves the structure of the semigroup and, in particular, sends generators to generators. This leaves only three possibilities for an isomorphism or anti-isomorphism between two semigroups arising from different choices for uv, vu, and v^2. They are the anti-isomorphism induced by the trivial permutation, and the isomorphism and anti-isomorphism induced by the transposition of u and v. As it is known that u^2 is the unique element in $S^2 \setminus S^3$ the latter two cases cannot occur if $v^2 \neq u^2$.

Case 1: $uv \in \{u^{n-2}, u^{n-1}\}$. Let $l \in \{2, 3, \cdots, n-1\}$ such that $vu = u^l$. From $(uv)u = u(vu) = u^{l+1}$ and the fact that $(uv)u$ equals the zero element, u^{n-1}, it follows that $l \in \{n-2, n-1\}$. In the same way $u^{m+1} = v(vu) = vu^l = u^{2l-1} = u^{n-1}$, if $v^2 = u^m$, and hence $m \in \{n-2, n-1\}$. Since all products of three elements involving v equal the zero element, u^{n-1}, the multiplication is associative. The 2 choices for each of uv, vu, and v^2 result in the 8 combinations illustrated in Figure 2.1. The multiplications $*_2$ and $*_3$ as well as $*_6$ and $*_7$ define pairs of

$*_1$	u	v
u	u^2	u^{n-2}
v	u^{n-2}	u^{n-2}

$*_2$	u	v
u	u^2	u^{n-1}
v	u^{n-2}	u^{n-2}

$*_3$	u	v
u	u^2	u^{n-2}
v	u^{n-1}	u^{n-2}

$*_4$	u	v
u	u^2	u^{n-1}
v	u^{n-1}	u^{n-2}

$*_5$	u	v
u	u^2	u^{n-2}
v	u^{n-2}	u^{n-1}

$*_6$	u	v
u	u^2	u^{n-1}
v	u^{n-2}	u^{n-1}

$*_7$	u	v
u	u^2	u^{n-2}
v	u^{n-1}	u^{n-1}

$*_8$	u	v
u	u^2	u^{n-1}
v	u^{n-1}	u^{n-1}

Figure 2.1 Multiplication of generators in semigroups with nilpotency rank $n-1$

dual semigroups. The transposition of u and v does not yield an isomorphism or anti-isomorphism since $v^2 \neq u^2$. Hence there are 6 non-equivalent semigroups in this case, 4 of which are commutative.

Case 2: $uv = u^k$ with $2 \leq k < n/2$. Let $l \in \{2, 3, \cdots, n-1\}$ such that $vu = u^l$. From $u^{k+1} = uvu = u^{l+1}$ it follows that $k = l$ since $k \leq n-3$ and the powers $u^3, u^4, \cdots, u^n$ for possible values of l are all different, except $u^{n-1} = u^n$. Using the same type of argument $v^2 = u^m$ implies $u^{m+1} = v^2u = vu^k = u^{2k-1}$, and hence $m = 2k-2$ since $k < n/2$. With $uv = vu = u^k$ and $v^2 = u^{2k-2}$ it follows that the value of any product in S is determined by how many times u and v appear. (A product containing i times u and j times v equals $u^{i+j(k-1)}$.) This makes the multiplication associative. Hence for every choice of k there is exactly one semigroup giving a total of $\lceil n/2 \rceil - 2$. All these semigroups are commutative and no two are equivalent since v^2 is different in every case. (In the case $k = 2$, where u and v are interchangeable, their transposition induces an automorphism.)

Case 3: $uv = u^k$ with $n/2 \leq k \leq n-3$. As in the previous case $vu = uv = u^k$. Now $vvu = u^{2k-1}$ equals the zero u^{n-1}. This leaves the two choices u^{n-2} and u^{n-1} for v^2. Similar to the previous case the value of a product only depends on the number of times u and v appear, making the multiplication associative. (In fact, every product with more than one v equals the zero.) Thus this case yields $2(n-2-\lceil n/2 \rceil)$ non-equivalent semigroups, all commutative.

No two semigroups from different cases are equivalent, which leads to a total of

$$6 + (\lceil n/2 \rceil - 2) + 2(n-2-\lceil n/2 \rceil) = n + \lfloor n/2 \rfloor$$

semigroups, all but 2 from the first case commutative. □

The next step is to consider nilpotent semigroups with n elements and nilpotency rank $n-2$. Due to Lemmas 2.1.2(i) and 2.1.5 the minimal generating sets of such semigroups contain either 2 or 3 elements.

Theorem 2.1.9 *Let S be a nilpotent semigroup of order n ($n \neq 5$) and nilpotency rank $n-2$. If $n \neq 6$ or S has a minimal generating set of size 3, then $S = T \cup \{x, y\}$ where T is a monogenic subsemigroup of S with nilpotency rank $n-2$.*

Proof: The statement is trivially true for $n \leq 4$.

Let $n \geq 6$ and let first $S = \langle u, v, w \rangle$ have a minimal generating set of size 3. Similar to the proof of Theorem 2.1.7 denote with t_k the unique element in $S^k \backslash S^{k+1}$ for $2 \leq k \leq n-3$. Considering a product $s_1 s_2 s_3 = t_3$ it follows with Lemma 2.1.2(ii) that s_1, s_2, and s_3 are generators and that $s_1 s_2 = s_2 s_3 = t_2$. Hence $s_1 s_1 s_2 = t_3$ and $s_1 s_1 = t_2$. Without loss of generality let $u^2 = t_2$. With the same arguments as in the proof of Theorem 2.1.7 it follows that $u^k = t_k$ for $2 \leq k \leq n-2$. In particular $|\langle u \rangle| = n-2$ and u^{n-2} is the zero element in S. Choosing $x = v$ and $y = w$ completes this part of the proof.

Let now $n \geq 7$ and $S = \langle u, v \rangle$. According to Corollary 2.1.4(i) the set $S^2 \backslash S^3$ must have size 2 and all $S^k \backslash S^{k+1}$ for $3 \leq k \leq n-3$ are of size 1. Let $s \in S^4 \backslash S^5$ and $s = s_1 s_2 s_3 s_4$. Without loss of generality $s_1 = u$. According to Lemma 2.1.2 the products $s_1 s_2 s_3$ and $s_2 s_3 s_4$ are in $S^3 \backslash S^4$, which contains only one element. Using this argument repeatedly yields

$$s_1s_2s_3s_4 = us_2s_3s_4 = uus_2s_3 = uuus_2 = u^4.$$

This means $\{u^k\} = S^k \backslash S^{k+1}$ for all $3 \leq k \leq n-3$ and $|\langle u \rangle| = n-2$. Choosing $x = v$ and y to equal the element in $S^2 \setminus S^3$ which does not equal u^2 completes the proof. □

For the classification of nilpotent semigroups with n elements and nilpotency rank $n-2$ the two cases depending on the size of the minimal generating set are treated separately in the next two theorems.

Theorem 2.1.10 *For $n \geq 6$ the number of nilpotent semigroups with n elements, nilpotency rank $n-2$ and minimal generating set of size 3 equals*

$$\begin{array}{ll} \frac{1}{8}(21n^2 + 22n - 96) & \textit{if } n \textit{ is even, and} \\ \frac{1}{8}(21n^2 + 36n - 81) & \textit{if } n \textit{ is odd.} \end{array}$$

Proof: Let $S = \langle u, v, w \rangle$ and $t_k = u^k \in S^k \setminus S^{k+1}$ for $2 \leq k \leq n-3$ as in the first part of the proof of the previous theorem. The proof to obtain the formula for the number of such semigroups S follows a similar approach as the one for Theorem 2.1.8. All of the products $uv, vu, v^2, uw, wu, w^2, vw$, and wv are in the set $S^2 = \{u^k \mid 2 \leq k \leq n-2\}$, and knowing them uniquely determines S. The different choices are discussed below in several cases. For each of the multiplications the value of a product with three factors will depend only on the number of times v and w appear, making all multiplications associative. Potential isomorphisms or anti-isomorphisms between semigroups are induced by permutations of the generators u, v, and w.

Case 1: $u^2 = v^2$. From $t_3 = u^2u = v^2u = vvu$ it follows that $vu = t_2 = u^2$. Analogously $uv = u^2$, leading to the situation illustrated in Figure 2.2. This case is divided further.

Case 1.a: $uw = u^k$ with $2 \leq k \leq n/2 - 1$. This case is similar to Case 2 in the proof of Theorem 2.1.8. It follows from $uvw = uuw = u^{k+1}$ that $vw = u^k$ and similarly $wu = wv = u^k$. From $uww = u^kw = u^{2k-1}$ one deduces that $w^2 = u^{2k-2}$. As the value of a product depends only on the number of times u, v, and w appear,

	u	v	w
u	u^2	u^2	$\diamond$
v	u^2	u^2	$\diamond$
w	$\diamond$	$\diamond$	$\diamond$

Figure 2.2 Multiplication of generators; $\diamond$ denotes undefined products

the multiplication is associative for each choice of k. This leads to

$$\lfloor n/2 \rfloor - 2$$

commutative semigroups. All of these are non-equivalent, as $u^2 \neq w^2 \neq v^2$ and the transposition of u and v induces an automorphism for each such semigroup.

Case 1.b: $uw = u^k$ with $n/2 - 1 < k \leq n - 4$. This case is similar to Case 3 in the proof of Theorem 2.1.8. As in the previous case $vw = wu = wv = u^k$, but now $uww = u^{2k-1}$ equals the zero u^{n-2} which leaves the two choices u^{n-3} and u^{n-2} for w^2. In any case an associative multiplication is defined, giving rise to

$$2(n - \lfloor n/2 \rfloor - 3)$$

semigroups. As before these are all commutative and pairwise non-equivalent.

Case 1.c: $uw \in \{u^{n-3}, u^{n-2}\}$. This case is similar to Case 1 in the proof of Theorem 2.1.8. All of wu, vw, wv, and w^2 take values in $\{u^{n-3}, u^{n-2}\}$. For all choices any product of three elements involving w equals the zero element and any other product of length three equals u^3. Hence, all of the $2^5 = 32$ choices lead to associative multiplications, many of which define equivalent semigroups. This becomes easier to see when considering the two subsemigroups $U = \langle u, w\rangle$ and $V = \langle v, w\rangle$ of S. Note that $u^k = v^k$ for all $2 \leq k \leq n-2$ and both U and V are nilpotent semigroups with $n-1$ elements and rank $n-2$. Fixing $w^2 = u^{n-2}$ leaves three choices for each of U and V according to the proof of Theorem 2.1.8. Exactly one of the choices is a non-commutative semigroup. Now, U and V can be interchanged by permuting u and v, since $u^2 = v^2$. This yields 6 different, unordered pairs (U, V). Each pair defines S up to equivalence, except when both U and V are non-commutative, in which case there are the two possibilities illustrated in Figure 2.3. This yields a total of 7 semigroups with $w^2 = u^{n-2}$, and another 7 if $w^2 = u^{n-3}$ using analogous reasoning,[1] which gives

$$14$$

semigroups from this case.

	u	v	w
u	u^2	u^2	u^{n-2}
v	u^2	u^2	u^{n-2}
w	u^{n-3}	u^{n-3}	u^{n-2}

	u	v	w
u	u^2	u^2	u^{n-3}
v	u^2	u^2	u^{n-2}
w	u^{n-2}	u^{n-3}	u^{n-2}

Figure 2.3 Multiplication of generators in semigroups with nilpotency rank $n-2$

Case 2: $v^2, w^2 \neq u^2$. Neither the transposition of u with v nor the transposition of u with w is an isomorphism or anti-isomorphism. Let $V = \langle u, v\rangle$ and $W = \langle u, w\rangle$. Both V and W are semigroups with $n-1$ elements and nilpotency rank $n-2$. According to the proof of Theorem 2.1.8 there are $n-1+\lfloor (n-1)/2\rfloor -1$ choices for V and W – taking into account that $v^2, w^2 \neq u^2$ – and all but two of the semigroups are commutative. This case is divided further.

Case 2.a: V, W not commutative. There are two choices for V and W, which can be either equivalent or not, and either $uv = uw$ or $vu = uw$. Taking into account that V and W can be interchanged by permuting v and w and after excluding duals, Figure 2.4 shows the remaining 6 constellations. No two of these will lead to equivalent semigroups. From $uvw = u^{n-2}$ it follows $vw, wv \in \{u^{n-3}, u^{n-2}\}$. For Tables I, III, V, and VI in Figure 2.4 all 4 ways to choose values for vw and wv lead to non-equivalent semigroups. For Tables II and IV, where V is equivalent to W and $uv = wu$, the semigroup where $vw = u^{n-3}$ and $wv = u^{n-2}$ is equivalent to the semigroup where $vw = u^{n-3}$ and $wv = u^{n-2}$ under the anti-isomorphism exchanging v and w. All together, this case leads to

$$4 \cdot 4 + 3 \cdot 2 = 22$$

non-equivalent semigroups S.

Case 2.b: V not commutative, W commutative. According to the proof of

[1]Alternatively one can examine all 16 possible choices after fixing $w^2 = u^{n-3}$ directly and identify isomorphic and anti-isomorphic ones.

I	u	v	w
u	u^2	u^{n-3}	u^{n-3}
v	u^{n-2}	u^{n-3}	$\diamond$
w	u^{n-2}	$\diamond$	u^{n-3}

II	u	v	w
u	u^2	u^{n-3}	u^{n-2}
v	u^{n-2}	u^{n-3}	$\diamond$
w	u^{n-3}	$\diamond$	u^{n-3}

III	u	v	w
u	u^2	u^{n-3}	u^{n-3}
v	u^{n-2}	u^{n-2}	$\diamond$
w	u^{n-2}	$\diamond$	u^{n-2}

IV	u	v	w
u	u^2	u^{n-3}	u^{n-2}
v	u^{n-2}	u^{n-2}	$\diamond$
w	u^{n-3}	$\diamond$	u^{n-2}

V	u	v	w
u	u^2	u^{n-3}	u^{n-3}
v	u^{n-2}	u^{n-2}	$\diamond$
w	u^{n-2}	$\diamond$	u^{n-3}

VI	u	v	w
u	u^2	u^{n-3}	u^{n-2}
v	u^{n-2}	u^{n-2}	$\diamond$
w	u^{n-3}	$\diamond$	u^{n-3}

Figure 2.4 Multiplication of generators; $\diamond$ denotes undefined products

Theorem 2.1.8 there are $2(n-1+\lfloor (n-1)/2 \rfloor - 3)$ combinations for V and W (the number of commutative semigroups excluding the one semigroup where both generators equal u^2). Again, from $uvw = u^{n-2}$ it follows $vw, wv \in \{u^{n-3}, u^{n-2}\}$ and all 4 choices lead to non-equivalent semigroups as v and w are not interchangeable. This leads to a total of

$$8\left(n-1+\left\lfloor \frac{n-1}{2} \right\rfloor - 3\right).$$

Case 2.c: V commutative, W not commutative. This case yields semigroups equivalent to those from the previous case.

Case 2.d: V, W commutative. Let $uv = u^k$ and $uw = u^l$. Then it follows from $uvw = u^{k+l-1}$ that $vw, wv \in \{u^{n-3}, u^{n-2}\}$ if $k + l \geq n-1$. Each case gives three non-equivalent semigroups, since the two choices, where vw and wv differ lead to two anti-isomorphic semigroups. If $k+l \leq n-2$ then $vw = wv = u^{k+l-2}$. Remember from the proof of Theorem 2.1.8 that there is one semigroup with $uv = u^k$ if $k < (n-1)/2$ and two otherwise. The respective statement holds as well for w replacing v. Thus out of the

$$\sum_{i=1}^{n-1+\lfloor \frac{n-1}{2} \rfloor - 3} i = \frac{1}{2}\left(\left(n+\left\lfloor \frac{n-1}{2} \right\rfloor\right)^2 - 7\left(n+\left\lfloor \frac{n-1}{2} \right\rfloor\right)\right) + 6$$

combinations of V and W there are

$$\sum_{k=3}^{\lceil\frac{n-1}{2}\rceil+1}\left(\sum_{l=k}^{n-k-2}1+\sum_{l=\lceil\frac{n-1}{2}\rceil}^{n-k-2}1\right)=\sum_{k=3}^{\lceil\frac{n-1}{2}\rceil+1}(n-2k-1)+\left(n-\left\lceil\frac{n-1}{2}\right\rceil-k-1\right)$$

that allow 1 solution and the others allow 3.

Summing all cases, evaluating the sums and simplifying the expression separately for even and odd integers yields the stated formula. □

Theorem 2.1.11 *For $n \geq 7$ there are $5n+\lfloor n/2\rfloor-\lceil n/3\rceil-1$ nilpotent semigroups with n elements, nilpotency rank $n-2$ and minimal generating set of size 2.*

Proof: The proof is similar to the ones for Theorems 2.1.8 and 2.1.10. As in the second part of the proof of Theorem 2.1.9 let $S = \langle u, v\rangle$ and denote by y the element in S, which is neither v nor a power of u. Since $y \in S^2 \setminus S^3$ at least one of uv, vu, v^2 has to equal y. Note that the values of these products with two generators together with the values of vu^2, vy, uy uniquely define S.[2] In the following different choices for the products are considered, depending – on the first level – on the products equalling y. All multiplications will again be associative, because the value of a product with three elements will only depend on the number of times v and y appear. Potential isomorphisms and anti-isomorphisms are induced by permutations of the generators u and v.

Case 1: $y = v^2 = uv = vu$. Let $vy = u^k$. Since u and v commute, this yields $uy = uv^2 = vuv = vy = u^k$ and $vu^2 = uvu = uy = u^k$. Then

$$u^{k+1} = uvy = uyv = u^kv = u^{k-1}uv = u^{k-1}y = u^{k-2}u^k = u^{2k-2},$$

[2]The remaining products are deduced as follows: $vu^k = (vu^2)u^{k-2}$; $u^kv = u^{k-1}(uv)$;

$$yu^k = \begin{cases} vuu^k = (vu^2)u^{k-1} & \text{if } vu = y\\ v^2u^k = v(vu)u^{k-1} = (vu)u^{l+k+2} = u^{2l+k-2} & \text{if } vu = u^l, v^2 = y\\ uvu^k = u(vu)u^{k-1} = u^{l+k} & \text{if } vu = u^l, uv = y;\end{cases}$$

$$yv = \begin{cases} v^2v = vv^2 = vy & \text{if } v^2 = y\\ uvv = uu^l = u^{l+1} & \text{if } v^2 = u^l, uv = y\\ vuv = vu^k = (vu^2)u^{k-2} & \text{if } uv = u^k, vu = y\end{cases}$$

.

which yields $k = 3$ or $k \geq n-3$, and thus leads to 3 non-equivalent semigroups.

Case 2: $y = vu = uv$. Let $v^2 = u^l$. It follows $vy = vvu = u^{l+1}$. Let $uy = u^k$ then $vu^2 = uvu = uy = u^k$ and

$$u^{l+2} = v^2u^2 = vuvu = vuy = vu^k = u^kv = u^{k-2}uy = u^{2k-2}. \qquad (2.1)$$

Case 2.a: $2 \leq l \leq n-5$. From (2.1) it follows $k = l/2 + 2$ which leads to a total of $\lceil n/2 \rceil - 3$ non-equivalent semigroups.

Case 2.b: $l \geq n-4$. This leaves 3 choices for l, and for each $k \geq n/2$ due to (2.1). Hence there are $3(\lfloor n/2 \rfloor - 1)$ non-equivalent semigroups from this case.

Case 3: $y = uv = v^2$. Let $vu = u^l$. Then

$$u^{l+1} = uu^l = uvu = yu = vvu = vu^l = vuu^{l-1} = u^{2l-1}$$

yields either $l = 2$ or $l \geq n-3$. Then $vu^2 = u^{l+1}$ and

$$uy = uv^2 = uvv = yv = vvv = vy = vuv = u^lv = u^{l-2}uy,$$

showing that all three values for l lead to valid choices for vu^2, uy, and vy. Hence this case accounts for 3 non-equivalent semigroups.

Case 3': $y = vu = v^2$. This case yields semigroups equivalent to those from Case 3.

Case 4: $v^2 = y$. Let $vu = u^k$ and $uv = u^l$. Then $u^{k+1} = uvu = u^{l+1}$ and hence $k = l$ or $k, l \in \{n-3, n-2\}$. Furthermore $vu^2 = vuu = u^{k+1}$ and $uy = uvv = u^lv = u^{2l-1}$. For the value of vy consider $uvy = u^lvv = u^{2l-1}v = u^{3l-2}$.

Case 4.a: $2 \leq l < n/3$. Then $vy = u^{3l-3}$ and $l < n-3$, which leads to $\lceil n/3 \rceil - 2$ non-equivalent semigroups.

Case 4.b: $n/3 \leq l \leq n-4$. Then $vy \in \{u^{n-3}, u^{n-2}\}$, leading in this case to $2(n-4-\lceil n/3 \rceil + 1)$ non-equivalent semigroups.

Case 4.c: $l \in \{n-3, n-2\}$. Again $vy \in \{u^{n-3}, u^{n-2}\}$. Recall that here $k \in \{n-3, n-2\}$. This leads to equivalent semigroups if one of k and l equals $n-3$ and the other equals $n-2$. The number of non-equivalent semigroups is 6.

Case 5: $vu = y$. Let $uv = u^k$ and $v^2 = u^l$. It follows $vy = vvu = u^{l+1}$ and $uy = uvu = u^{k+1}$. For vu^2 consider $uvu^2 = u^{k+2}$.

Case 5.a: $2 \leq k \leq n-5$. Here $vu^2 = u^{k+1}$. From $u^{l+1} = vvu = vu^k = u^{2k-1}$

it follows $l = 2k - 2$, or $k \geq (n-1)/2$ and $l \in \{n-3, n-2\}$. The former yields $\lceil (n-1)/2 \rceil - 2$ non-equivalent semigroups, and the latter $2(n - \lceil (n-1)/2 \rceil - 4)$.

Case 5.b: $n-4 \leq k \leq n-2$. Here $vu \in \{n-3, n-2\}$, and $l \in \{n-3, n-2\}$, leading to $3 * 2 * 2 = 12$ non-equivalent semigroups.

Case 5': $uv = y$. This case yields semigroups equivalent to those from Case 5.

In contrast to before there are two semigroups from different cases which are equivalent. The only possibility for this to happen is that the transposition of u and v induces an isomorphism (or anti-isomorphism). For this v has to generate a subsemigroup of size $n-2$, and hence $v^i = u^i$ for all $3 \leq i \leq n-2$. This occurs, if $k = 3$ in Case 1 and if $l = 2$ in Case 4. Therefore adding the numbers from all cases together and subtracting 1 yields the stated formula. □

Combining Theorems 2.1.10 and 2.1.11 gives the following result.

Corollary 2.1.12 *For $n \geq 7$ the number of nilpotent semigroups with n elements and nilpotency rank $n-2$ is*

$$\begin{array}{ll} \frac{1}{8}(21n^2 + 66n - 104) - \left\lceil \frac{n}{3} \right\rceil & \text{if } n \text{ is even, and} \\ \frac{1}{8}(21n^2 + 80n - 93) - \left\lceil \frac{n}{3} \right\rceil & \text{if } n \text{ is odd.} \end{array}$$

The case $n = 5$ is included in the forthcoming studies of 3-nilpotent semigroups in the next sections. No further considerations will be undertaken for $n = 6$ as the semigroups of this order are long known, see [Ple67]. There are 43 nilpotent semigroups with 6 elements, nilpotency rank 4, and 2 generators, and hence the number of 4-nilpotent semigroups with 6 elements is 142. They are available in the GAP [GAP08] package Smallsemi [DM10] which is explained in Section 4.2.

The presented method of classifying nilpotent semigroups by their rank reaches its limits at this point. The proofs of the formulae for the number of nilpotent semigroups S with n elements and rank $n-1$ or $n-2$ rely on the fact that $S^3 \setminus S^4$ contains only one element. This is no longer true for all semigroups of nilpotency rank $n-3$. Already to attempt a classification of nilpotent semigroups of order n and rank $n-3$ would seem to be a project in its own right.

Question 2.1.13 Can the methods in this section be extended to classify nilpotent semigroups S for which $|S^3 \setminus S^4| > 1$?

An additional problem is that the results and their proofs in this section indicate an increasing number of exceptions for small orders, when the nilpotency rank decreases. This would make any formula worthless for the enumeration of all semigroups of the smallest unknown order(s).

Alternatively the classification can be approached by increasing nilpotency rank. The results for 1-nilpotent and 2-nilpotent semigroups were mentioned in this section. Preparation for the far more challenging task of classifying semigroups with nilpotency rank 3 is compiled in the next section.

2.2 Power Group Enumeration

Given an action of a group G on a finite set X, what is the number of orbits X forms under G? Sophisticated methods to answer this – and more detailed – questions effectively in a large number of settings were first developed by Redfield [Red27] and Polya [Pol37]. In this section a modified version of a theorem by de Bruijn [dB59] is presented, which will be used in the next section to enumerate 3-nilpotent semigroups. Most of the content is based on [HP73], with slight adjustments, particularly in the notation.

It is a well-known result that the question starting this section can be answered by counting for each element $g \in G$ the number of points $x \in X$ with $x^g = x$. For a permutation $\pi \in S_X$ let $\delta(\pi, k)$ denote the number of cycles of length k in the disjoint cycle notation of π. Then the result reads as follows.

Lemma 2.2.1 (Cauchy-Frobenius) *Let G be a subgroup of S_X. The number of orbits of X under G is*

$$N(G) = \frac{1}{|G|} \sum_{g \in G} \delta(g, 1).$$

For two proofs of this lemma – together with enlightening information on the confusion about its name – see [Neu79].

Depending on the action – which might not be presented as a subgroup of S_X – it can be difficult to apply Lemma 2.2.1. A tool that at first seems to require even more knowledge will turn out very helpful for actions encountered later.

Definition 2.2.2 Let G be a subgroup of S_n. Then the polynomial

$$\mathcal{Z}(G; x_1, x_2, \ldots, x_n) = \frac{1}{|G|} \sum_{g \in G} \prod_{k=1}^{n} x_k^{\delta(g,k)}$$

is the *cycle index* of the group G, in short denoted by $\mathcal{Z}(G)$.

Consider the following example, copied from [HP73], of finding the cycle index of a specific group.

Example 2.2.3 Let G be S_3, the symmetric group on 3 elements. The identity permutation $(1)(2)(3)$ has three cycles of length 1, resulting in the term x_1^3. The three permutations $(1)(23), (2)(13)$, and $(3)(12)$ of order 2 each have one cycle of length 1 and one of length 2, together leading to the term $3x_1x_2$. Finally, the two permutations (123) and (132) contribute $2x_3$ to the cycle index. Thus it is

$$\mathcal{Z}(S_3) = \frac{1}{3!}(x_1^3 + 3x_1x_2 + 2x_3).$$

In the previous example the permutations are collected according to the structure of their disjoint cycle notation. Each collection, and thus each structure, is associated with a partition of 3. In general, if j is a partition of n, written as $j \vdash n$, then denote by j_i the number of summands equalling i. This yields $j_i = \delta(g, i)$ for each element g in the collection[3] associated with j. This observation allows one to write the cycle index of the symmetric group in a more compact form.

Lemma 2.2.4 *The cycle index of S_n is*

$$\mathcal{Z}(S_n) = \sum_{j \vdash n} \left(\prod_{i=1}^{n} j_i! i^{j_i} \right)^{-1} \prod_{a=1}^{n} x_a^{j_a}.$$

Proof: First it shall be shown that the number of elements in S_n with cycle decomposition according to a partition $j \vdash n$ equals

$$\frac{n!}{\prod_{i=1}^{n} j_i! i^{j_i}}. \tag{2.2}$$

[3] For S_n these 'collections' are actually conjugacy classes.

The partition fixes the structure of the cycle decomposition. There are $n!$ ways to distribute the n elements, but some ways result in the same group element. By rotation each cycle of length i allows i different ways to write it. Moreover, cycles of the same length are interchangeable. In total each permutation appears written in $\prod_{i=1}^{n} j_i! i^{j_i}$ different ways.

Equation (2.2), together with the definition of the cycle index, yields the formula when summing over partitions of n. □

It was shown in the mathematical background section how the natural action of S_n leads to a more complicated action on multiplication tables. Other actions, that are of interest in this thesis, build from existing actions in a similar way.

Definition 2.2.5 Let A and B be finite permutation groups acting on finite disjoint sets X respectively Y.

(i) The *sum group* $AB = \{\alpha\beta \mid \alpha \in A, \beta \in B\}$ acts on the union $X \cup Y$. Each $\alpha\beta \in AB$ acts like α on elements in X and like β on elements in Y.

(ii) The *power group* $B^A = \{(\alpha;\beta) \mid \alpha \in A, \beta \in B\}$ acts on Y^X, the set of functions from X to Y. The image of $f \in Y^X$ under $(\alpha;\beta)$ is given by

$$f^{(\alpha;\beta)}(x) = \left(f\left(x^{\alpha}\right)\right)^{\beta}.$$

Be aware that both B^A and AB are, as groups, isomorphic to the direct product $A \times B$. The notation in Definition 2.2.5 implicitly incorporates the action.

Lemma 2.2.6 *Let AB be a sum group. Then $\mathcal{Z}(AB) = \mathcal{Z}(A)\mathcal{Z}(B)$.*

Proof: The result follows immediately from Definition 2.2.2 of the cycle index and Definition 2.2.5 of the sum group. □

For the power group the full cycle index is not required in this thesis. Of interest is the constant form of the power group enumeration theorem, which yields the number of orbits under the action of a power group. As mentioned, the result goes back to de Bruijn [dB59], but is presented here in the form given in [HP73].

Theorem 2.2.7 *Let A, B, X, and Y be as in Definition 2.2.5. The number of orbits of functions in Y^X under the power group B^A equals*

$$\frac{1}{|B|}\sum_{\beta\in B}\mathcal{Z}(A; c_1(\beta), c_2(\beta), \dots, c_{|X|}(\beta)),$$

where

$$c_i(\beta) = \sum_{d|i} d\delta(\beta, d).$$

Proof: The proof is only outlined and is available in [HP73, Section 6.1] in full.

The number of functions stabilised by an element in the power group is

$$\delta((\alpha;\beta), 1) = \prod_{i=1}^{m}\left(\sum_{d|i} d\delta(\beta, d)\right)^{\delta(\alpha,i)}.$$

Applying Lemma 2.2.1 proves the statement. □

To apply Theorem 2.2.7 the cycle index of A has to be known. So far, this is the case for symmetric groups (Lemma 2.2.4) and hence sum groups of symmetric groups (Lemma 2.2.6). This knowledge is not sufficient for the forthcoming section. Three more group actions are considered and their cycle indices computed for special cases.

Definition 2.2.8 Let A be a group acting on X.

(i) The group acting on $X^k = \{(x_1, x_2, \dots, x_k) \mid x_i \in X\}$, the k-fold Cartesian product, componentwise like A (that is, $(x_1, x_2, \dots, x_k)^\alpha = (x_1^\alpha, x_2^\alpha, \dots, x_k^\alpha)$ for $\alpha \in A$) is denoted by $A^{\times k}$.

(ii) The group acting on the set $X^{\{k\}} = \{\{x_1, x_2, \dots, x_k\} \mid x_i \in X\}$ containing all subsets of X with at most k elements pointwise like A (that is, $\{x_1, x_2, \dots, x_k\}^\alpha = \{x_1^\alpha, x_2^\alpha, \dots, x_k^\alpha\}$ for $\alpha \in A$) is denoted by $A^{\{k\}}$.

(iii) Denote by $2A^{\times 2}$ the group with elements $\{\alpha, \bar{\alpha} \mid \alpha \in A\}$ acting on X^2 as follows. For an element of the form $\alpha \in A$ the image of (x_1, x_2) is (x_1^α, x_2^α) as in $A^{\times 2}$, and $(x_1, x_2)^{\bar{\alpha}} = (x_2^\alpha, x_1^\alpha)$.

The cycle indices of the groups $A^{\times k}$, $A^{\{k\}}$, and $2A^{\times 2}$ can be deduced if one knows the cycle index of the underlying group A. To do this for the special case when $A = S_n$ and $k = 2$ completes the preparation for the next section.

Lemma 2.2.9 *Let $n \in \mathbb{N}$.*

(i) *The cycle index of $S_n^{\times 2}$ is*

$$\mathcal{Z}(S_n^{\times 2}) = \sum_{j \vdash n} \left(\prod_{i=1}^{n} j_i!\, i^{j_i} \right)^{-1} \prod_{a,b=1}^{n} x_{\operatorname{lcm}(a,b)}^{j_a j_b \gcd(a,b)}. \tag{2.3}$$

(ii) *The cycle index of $2S_n^{\times 2}$ is*

$$\mathcal{Z}(2S_n^{\times 2}) = \frac{1}{2}\mathcal{Z}(S_n^{\times 2}) + \frac{1}{2} \sum_{j \vdash n} \left(\prod_{i=1}^{n} j_i!\, i^{j_i} \right)^{-1} \prod_{a=1}^{n} \left(q_a^{j_a} p_{a,a}^{j_a^2 - j_a} \prod_{b=1}^{a-1} p_{a,b}^{2 j_a j_b} \right), \tag{2.4}$$

where the monomials are $p_{a,b} = x_{\operatorname{lcm}(2,a,b)}^{ab/\operatorname{lcm}(2,a,b)}$ and

$$q_a = \begin{cases} x_a x_{2a}^{(a-1)/2} & \text{if } a \equiv 1 \mod 2 \\ x_a^a & \text{if } a \equiv 0 \mod 4 \\ x_{a/2}^2 x_a^{a-1} & \text{if } a \equiv 2 \mod 4. \end{cases}$$

(iii) *The cycle index of $S_n^{\{2\}}$ is*

$$\mathcal{Z}(S_n^{\{2\}}) = \sum_{j \vdash n} \left(\prod_{i=1}^{n} j_i! i^{j_i} \right)^{-1} \prod_{a=1}^{\lfloor n/2 \rfloor} r_a \prod_{a=1}^{\lfloor (n+1)/2 \rfloor} s_a \prod_{a=1}^{n} t_a \left(\prod_{b=1}^{a-1} x_{\operatorname{lcm}(a,b)}^{j_a j_b \gcd(a,b)} \right), \tag{2.5}$$

where the monomials are $r_a = x_a^{j_{2a}} x_{2a}^{a j_{2a}}$, $s_a = x_{2a-1}^{a j_{2a-1}}$, and $t_a = x_a^{a(j_a^2 - j_a)/2}$.

Proof: (i): By definition each permutation in S_n induces a permutation in $S_n^{\times 2}$. Let $\alpha \in S_n$ and let z_a and z_b be two cycles thereof with length a and b respectively. Consider the action of α on those pairs in $[n]^2$ which have as first component an element in z_a and as second component an element in z_b. Let $(i, j) \in [n]^2$ be one such pair. Since $i^{\alpha^k} = i$ if and only if $a \mid k$, and $j^{\alpha^k} = j$ if and only if $b \mid k$, the pair (i, j) is in an orbit of length $\operatorname{lcm}(a, b)$. The total number of pairs with first component in z_a and second component in z_b equals ab. Hence there are

$\gcd(a,b)$ orbits. Repeating this consideration for every pair of cycles in α leads to $\prod_{a,b=1}^{n} x_{\operatorname{lcm}(a,b)}^{\delta(\alpha,a)\delta(\alpha,b)\gcd(a,b)}$ as contribution of the permutation induced by α to the cycle index $\mathcal{Z}(S_n^{\times 2})$. This yields

$$\mathcal{Z}(S_n^{\times 2}) = \frac{1}{n!} \sum_{\alpha \in S_n} \prod_{a,b=1}^{n} x_{\operatorname{lcm}(a,b)}^{\delta(\alpha,a)\delta(\alpha,b)\gcd(a,b)}.$$

That the contribution of α only depends on its cycle structure allows one to replace the summation over all group elements by a summation over partitions of n. The number of elements with cycle structure associated to a partition $j \vdash n$ is known from the proof of Lemma 2.2.4. Therefore

$$\mathcal{Z}(S_n^{\times 2}) = \frac{1}{n!} \sum_{j \vdash n} \frac{n!}{\prod_{i=1}^{n} j_i! \, i^{j_i}} \prod_{a,b=1}^{n} x_{\operatorname{lcm}(a,b)}^{j_a j_b \gcd(a,b)},$$

and cancelling the factor $n!$ proves (2.3).

(ii): For group elements in $2S_n^{\times 2}$ that are in $S_n^{\times 2}$ the contribution to the cycle index has just been computed. It is rearranged as follows to illustrate which contributions come from identical cycles and which from disjoint cycles:

$$\prod_{a,b=1}^{n} x_{\operatorname{lcm}(a,b)}^{\delta(\alpha,a)\delta(\alpha,b)\gcd(a,b)} = \prod_{a=1}^{n} \left(x_a^{a\delta(\alpha,a)} x_a^{a(\delta(\alpha,a)^2 - \delta(\alpha,a))} \prod_{b<a} x_{\operatorname{lcm}(a,b)}^{2\delta(\alpha,a)\delta(\alpha,b)\gcd(a,b)} \right).$$

For group elements of the form $\bar{\alpha}$ the contribution is going to be deduced from the one of α. Let z_a and z_b again be two cycles in α of length a and b respectively, and assume at first, they are disjoint. Then z_a and z_b induce $2\gcd(a,b)$ orbits of length $\operatorname{lcm}(a,b)$ on the $2ab$ elements in $[n]^2$ with one component from each of the two cycles. Let

$$\omega = \left\{ (i_1, j_1), (i_2, j_2), \ldots, (i_{\operatorname{lcm}(a,b)}, j_{\operatorname{lcm}(a,b)}) \right\} \tag{2.6}$$

be such an orbit. Then

$$\bar{\omega} = \left\{ (j_1, i_1), (j_2, i_2), \ldots, (j_{\operatorname{lcm}(a,b)}, i_{\operatorname{lcm}(a,b)}) \right\} \tag{2.7}$$

is another one. The set $\omega \cup \bar{\omega}$ is closed under the action of $\bar{\alpha}$. In how many orbits $\omega \cup \bar{\omega}$ splits depends on the parity of a and b. Acting with $\bar{\alpha}$ on (i_1, j_1) for $\operatorname{lcm}(a,b)$

times gives (i_1, j_1) if $\operatorname{lcm}(a,b)$ is even and (j_1, i_1) if $\operatorname{lcm}(a,b)$ is odd. Hence the two orbits ω and $\bar{\omega}$ merge to one orbit in the latter case and give two new orbits of the original length otherwise. This yields the monomial

$$x_{\operatorname{lcm}(2,a,b)}^{2ab/\operatorname{lcm}(2,a,b)} = \begin{cases} x_{\operatorname{lcm}(a,b)}^{2\gcd(a,b)} & \text{if} \quad \operatorname{lcm}(a,b) \equiv 0 \bmod 2 \\ x_{2\operatorname{lcm}(a,b)}^{\gcd(a,b)} & \text{if} \quad \operatorname{lcm}(a,b) \equiv 1 \bmod 2, \end{cases}$$

which appears $\delta(\alpha,a)\delta(\alpha,b)$ times if $a \neq b$ and $(\delta(\alpha,a)^2 - \delta(\alpha,a))/2$ times if $a = b$.

Let z_a and z_b now be identical and equal to the cycle $(i_1 i_2 \cdots i_a)$. The contribution to the monomial of α is the factor x_a^a. The orbits are of the form $\{(i_j, i_h) \mid 1 \le j, h \le a, j \equiv h + s \bmod a\}$ for $0 \le s \le a-1$. For $s = 0$ the orbit consists of pairs with equal entries, that is, $\{(i_1, i_1), (i_2, i_2) \dots (i_a, i_a)\}$, and thus stays the same under $\bar{\alpha}$. For an orbit $\omega = \{(i_j, i_h) \mid 1 \le j, h \le a, j \equiv h + s \bmod a\}$ with $s \neq 0$ define $\bar{\omega}$ as in (2.7). If $\omega \neq \bar{\omega}$ one argues like in the case of two disjoint cycles and gets the result depending on the parity of a. Note that $\omega = \bar{\omega}$ if and only if $s = a/2$. In particular this does not occur for a odd in which case

$$x_a x_{2a}^{(a-1)/2}$$

is the factor contributed to the monomial of $\bar{\alpha}$. If on the other hand a is even, one more case split is needed to deal with the orbit

$$\omega = \{(i_j, i_h) \mid 1 \le j, h \le a, j \equiv h + a/2 \bmod a\}.$$

Acting with $\bar{\alpha}$ on $(i_a, i_{a/2})$ for $a/2$ times gives $(i_a, i_{a/2})$ if $a/2$ is odd and $(i_{a/2}, i_a)$ if $a/2$ is even. Thus ω splits into two orbits of length $a/2$ in the former case and stays one orbit in the latter. The resulting factors contributed to the monomial of $\bar{\alpha}$ are therefore

$$\begin{array}{rlll} x_a^a & \text{if} & a \equiv 0 & \bmod 4 \\ x_{a/2}^2 x_a^{a-1} & \text{if} & a \equiv 2 & \bmod 4. \end{array}$$

Following the analysis for all pairs of cycles in α leads to the contribution of $\bar{\alpha}$ to the cycle index. Summing as before over all different partitions of n, which correspond to the different cycle structures, proves (2.4) as formula for $\mathcal{Z}(2S_n^{\times 2})$.

(iii): To compute $\mathcal{Z}(S_n^{\{2\}})$ let ω and $\bar{\omega}$ as in (2.6) and (2.7) be orbits for two cycles z_a and z_b from $\alpha \in S_n$ acting on X^2. If the two cycles z_a and z_b are disjoint then both ω and $\bar{\omega}$ correspond to the same orbit

$$\big\{\{i_1, j_1\}, \{i_2, j_2\}, \ldots, \{i_{\mathrm{lcm}(a,b)}, j_{\mathrm{lcm}(a,b)}\}\big\}$$

of α acting on $X^{\{2\}}$. The contribution to the monomial of α is therefore $x_{\mathrm{lcm}(a,b)}^{\gcd(a,b)}$. Let z_a and z_b now be identical and equal to the cycle $(i_1 i_2 \cdots i_a)$. In $S_n^{\times 2}$ this gave rise to the orbits $\{(i_j, i_h) \mid 1 \leq j, h \leq a, j \equiv h + s \bmod a\}$ for $0 \leq s \leq a-1$. The corresponding orbit under $S_n^{\{2\}}$ for $s = 0$ becomes $\{\{i_1\}, \{i_2\}, \ldots, \{i_a\}\}$. All other orbits become $\{\{i_j, i_h\} \mid 1 \leq j, h \leq a, j \equiv h + s \bmod a\}$ in the same way as before, but these are identical for s and $a - s$. This yields one further exception if a is even and $s = a/2$, in which case the orbit collapses to $\{\{i_j, i_{j+a/2}\} \mid 1 \leq j \leq a/2\}$. In total, identical cycles lead to the monomials

$$\begin{array}{rll} x_{a/2} x_a^{a/2} & \text{if} & a \equiv 0 \mod 2 \\ x_a^{(a+1)/2} & \text{if} & a \equiv 1 \mod 2. \end{array}$$

Summing once more over conjugacy classes and making the case split depending on the parity proves the formula for $\mathcal{Z}(S_n^{\{2\}})$. □

Formulae like those in the previous lemma for slightly different actions are given in [HP73, (4.1.9)] and [HP73, (5.1.5)]. The proof techniques used here are essentially the same as in [HP73]. A reference for the cycle index of the groups in Lemma 2.2.9 is not known to the author of this thesis.

2.3 3-nilpotent Semigroups

As mentioned in the preface, Kleitman, Rothschild, and Spencer asymptotically counted the number of different semigroups on an n element set [KRS76]. The cardinality of Z_n, the set of different 3-nilpotent semigroups on $[n]$, is identified as an asymptotic lower bound,[4] though part of the proof is only outlined. The latter influenced Jürgensen, Milgi, and Szék in [JMS91] to give $|Z_n|$ as a mere lower bound for the number of different semigroups on $[n]$. At the same time they suspect

[4] Strictly speaking only a subset of all 3-nilpotent semigroups is counted

$(1/2n!)|Z_n|$ to be a good lower bound for the number of non-equivalent semigroups with n elements. This belief was supported by the numbers for semigroups of order 7 and later backed up by the analysis in [SYT94, Section 8] for semigroups of order 8.

In this section the construction of 3-nilpotent semigroups from [KRS76], together with the enumeration techniques presented in the previous section, is used to establish an exact formula for the numbers $|\widehat{Z_n}|$ and $|\overline{Z_n}|$ of 3-nilpotent semigroups up to isomorphism and up to equivalence. To employ the construction found in [KRS76] for the counting, it is made more precise: for $n \geq 2$ let B be a subset of $[n]$ with $1 \leq |B| \leq n-1$ and let A be the complement of B in $[n]$. If $z \in B$ and $\psi : A \times A \to B$ is any function, then define a magma $S(\psi, z)$ on $[n]$ with multiplication as follows

$$xy = \begin{cases} \psi(x,y) & x, y \in A \\ z & \text{otherwise.} \end{cases}$$

It is easy to verify that any product abc in $S(\psi, z)$ equals z, meaning that the multiplication is associative and $S(\psi, z)$ forms a semigroup of nilpotency rank at most 3. The semigroup $S(\psi, z)$ is 2-nilpotent – and hence a zero semigroup – if and only if ψ is the constant function with value z. Conversely, choosing $B = S^2$ for a 3-nilpotent semigroup S on $[n]$, shows that S can be constructed as described above. Before moving on to the enumeration up to isomorphism, the given construction is used to count all different 3-nilpotent semigroups on $[n]$.

Theorem 2.3.1 *For $n \in \mathbb{N}$ the number of different 3-nilpotent semigroups on $[n]$ equals*

$$|Z_n| = \sum_{m=2}^{m_0(n)} \binom{n}{m} m \sum_{i=0}^{m-1} (-1)^i \binom{m-1}{i} (m-i)^{(n-m)^2},$$

where

$$m_0(n) = \left\lfloor n + 1/2 - \sqrt{n - 3/4} \right\rfloor .$$

Proof: For a fixed size m of B there are $\binom{n}{m}$ choices for B and then m choices for z. The number of functions from A^2 to B is $m^{(n-m)^2}$. To avoid counting semigroups twice for different m, only those functions for which each element in $B \setminus \{z\}$ appears as image shall be counted. For a subset of $B \setminus \{z\}$ of size i there are $(m-i)^{(n-m)^2}$ functions not having any of the elements in the subset as image. The inclusion-exclusion principle yields

$$\sum_{i=0}^{m-1}(-1)^i\binom{m-1}{i}(m-i)^{(n-m)^2} \tag{2.8}$$

as number of functions with $B \setminus \{z\}$ in their image.

The function ψ is defined on a set with $(n-m)^2$ elements. Hence, the condition that every element in $B \setminus \{z\}$ is an image yields the inequality $m-1 \leq (n-m)^2$. Reformulation gives $m \leq n + 1/2 - \sqrt{n-3/4}$. Summing (2.8) over all values for which 3-nilpotent semigroups occur proves the lemma. $\square$

In [JMS91, Theorem 15.3] the formula in the previous lemma is given as the number of a proper subclass of the semigroups in Z_n, while their proof shows that the exact number of different 3-nilpotent semigroups on $[n]$ is counted. Note that in agreement with the results from Section 2.1 the formula yields $|Z_1| = |Z_2| = 0$ and $|Z_3| = 1$.[5]

The more complicated counting of non-isomorphic and non-equivalent semigroups in Z_n is done in several steps. In a first step semigroups with 1 as zero element are considered and, as in the proof of the previous lemma, they are distinguished by the number of different entries in their multiplication table. For $n \geq 3$ and $2 \leq m \leq n-1$ define

$$Z_n(m) = \left\{S(\psi, 1) \mid \psi : ([n] \setminus [m])^2 \to [m] \text{ with } [m] \setminus [1] \subseteq \operatorname{im}(\psi)\right\}.$$

None of these sets contains a zero semigroup, as $\operatorname{im}(\psi)$ consists of the zero element for such a semigroup.

Lemma 2.3.2 *Let S be a 3-nilpotent semigroup with n elements. Then*

$$m \in \left\{2, \ldots, \left\lfloor n + 1/2 - \sqrt{n-3/4}\right\rfloor\right\}$$

given by $m = |S^2|$ is unique such that there exists a semigroup in $Z_n(m)$ equivalent to S.

[5] The convention $0^0 = 1$ is used.

Proof: As each element in S^2, other than the zero element, is a product of two generators, it follows that $2 \leq |S^2| \leq 1 + |S \setminus S^2|^2$. This leads to the inequality $m \leq 1 + (n-m)^2$ which yields $m \leq n + 1/2 - \sqrt{n - 3/4}$.

Let z denote the zero element of S, and let $f : S \to [n]$ be any bijection such that $f(z) = 1$ and $f(S^2) = [m]$. Then define $\psi : ([n] \setminus [m])^2 \to [m]$ by

$$\psi(i, j) = f(f^{-1}(i)f^{-1}(j)).$$

Now, since S is 3-nilpotent, if $x \in [m] \setminus [1]$, there exist $s, t \in S \setminus S^2$ such that $f(st) = x$. Thus $\psi(f(s), f(t)) = x$ and $[m] \setminus [1] \subseteq \mathrm{im}(\psi)$. Hence $S(\psi, 1) \in Z_n(m)$ and it remains to show that f is an isomorphism. If $x, y \in S \setminus S^2$, then $f(x)f(y) = \psi(f(x), f(y)) = f(xy)$. Otherwise, $x \in S^2$ or $y \in S^2$, in which case $f(x)f(y) = 1 = f(z) = f(xy)$.

The uniqueness of m follows from the fact that $|S^2|$ is preserved by isomorphism and anti-isomorphism, and $|S(\psi, 1)^2| = k$ for all $S(\psi, 1) \in Z_n(k)$. $\square$

Of course, in the previous lemma, it is not true in general that there exists a unique semigroup in $Z_n(m)$ equivalent to S, or, in other words, $|\overline{Z_n(m)}| < |Z_n(m)|$ in most cases. That each structural type of semigroup appears in exactly one of the sets $Z_n(m)$ allows one to determine the number of types of semigroups in each set independently. Together with Lemma 2.3.2 it follows in particular that

$$|\overline{Z_n}| = \sum_{m=2}^{m_0(n)} |\overline{Z_n(m)}|, \quad \text{where} \quad m_0(n) = \left\lfloor n + 1/2 - \sqrt{n - 3/4} \right\rfloor. \tag{2.9}$$

Before $|\overline{Z_n(m)}|$ is determined, the semigroups in $Z_n(m)$ are counted up to isomorphism; this being somewhat simpler but involving the same ideas as counting up to equivalence.

Isomorphisms between semigroups in $Z_n(m)$ induce an equivalence of functions from $([n] \setminus [m])^2$ into $[m]$, which define the semigroups. For $S(\psi, 1) \in Z_n(m)$ and $\pi \in S_n$ one notes that the semigroup $S(\psi, 1)^\pi$ lies in $Z_n(m)$ if and only if π stabilises $[n] \setminus [m]$ and $[1]$ – and hence $[m] \setminus [1]$ – setwise. Thus the actions on source and range of ψ are independent. The equivalence can then be captured using a power group action.

Lemma 2.3.3 *Two semigroups $S(\psi, 1)$ and $S(\chi, 1)$ in $Z_n(m)$ are isomorphic if and only if ψ and χ lie in the same orbit under the action of the power group $(S_{[n]\setminus[m]}^{\times 2})^{S_{[1]}S_{[m]\setminus[1]}}$.*

Proof: ($\Rightarrow$) By assumption there exists a $\pi \in S_n$ such that $\pi : S(\psi, 1) \to S(\chi, 1)$ is an isomorphism. As π stabilises $[m] \setminus [1]$ setwise and $1\pi = 1$, there do exist $\tau \in S_{[1]}S_{[m]\setminus[1]}$ and $\sigma \in S_{[n]\setminus[m]}$ induced by π. Then for all $x, y \in [n] \setminus [m]$

$$\psi(x, y) = (\psi(x, y)^\pi)^{\pi^{-1}} = (\chi(x^\pi, y^\pi))^{\pi^{-1}} = (\chi(x^\sigma, y^\sigma))^{\tau^{-1}} = \chi^{(\sigma;\tau^{-1})}(x, y).$$

($\Leftarrow$) Since ψ and χ lie in the same orbit under the action of the power group $(S_{[n]\setminus[m]}^{(2)})^{S_{[1]}S_{[m]\setminus[1]}}$, there exist $\sigma \in S_{[n]\setminus[m]}$ and $\tau \in S_{[1]}S_{[m]\setminus[1]}$ such that $\psi^{(\sigma;\tau)} = \chi$. Let $\pi \in S_{[n]}$ be defined by

$$x^\pi = \begin{cases} x^\sigma & x \in \{m+1, \ldots, n\} \\ x^{\tau^{-1}} & x \in \{1, \ldots, m\}. \end{cases}$$

It shall be shown that π is an isomorphism from $S(\psi, 1)$ to $S(\chi, 1)$: if $x, y \in [n]\setminus[m]$, then

$$x^\pi y^\pi = \psi(x^\sigma, y^\sigma) = (\psi(x^\sigma, y^\sigma)^\tau)^{\tau^{-1}} = (\psi^{(\sigma;\tau)}(x, y))^{\tau^{-1}} = (\chi(x, y))^{\tau^{-1}} = (xy)^\pi.$$

Otherwise, without loss of generality, suppose that $y \in \{1, \ldots, m\}$. Then

$$(xy)^\pi = 1^\pi = 1^{\tau^{-1}} = 1 = x^\sigma y^{\tau^{-1}} = x^\pi y^\pi,$$

as required. □

Lemma 2.3.3 shows that the number of non-isomorphic semigroups in $Z_n(m)$ equals the number of orbits of functions defining semigroups in $Z_n(m)$ under the power group action. The latter is obtained by application of Theorem 2.2.7.

Lemma 2.3.4 *For $p, q \in \mathbb{N}$ with $1 \leq q < p$ let $N(p, q)$ denote the number of orbits of functions from $([p] \setminus [q])^2$ into $[q]$ under the action of the power group*

$(S^{\times 2}_{[n]\setminus[m]})^{S_{[1]}S_{[m]\setminus[1]}}$. *Then*

$$N(p,q)=\sum_{j\vdash q-1}\sum_{k\vdash p-q}\left(\prod_{i=1}^{q-1}j_i!\,i^{j_i}\prod_{i=1}^{p-q}k_i!\,i^{k_i}\right)^{-1}\prod_{a,b=1}^{p-q}\left(1+\sum_{d\mid\operatorname{lcm}(a,b)}dj_d\right)^{k_ak_b\gcd(a,b)}.$$

Proof: Theorem 2.2.7 immediately yields that

$$N(p,q)=\frac{1}{(q-1)!}\sum_{\beta\in S_{[q]\setminus[1]}}\mathcal{Z}(S^{\times 2}_{[p]\setminus[q]};c_1(\beta),\ldots,c_{(p-q)^2}(\beta)),\tag{2.10}$$

where

$$c_i(\beta)=\sum_{d\mid i}d\delta(\beta,d).$$

As mentioned before $\mathcal{Z}(S^{\times 2}_{[p]\setminus[q]};c_1(\beta),\ldots,c_{(p-q)^2}(\beta))$ depends only on the cycle structure of β and is therefore an invariant of the conjugacy classes of $S_{[1]}S_{[q]\setminus[1]}$. Then again, these can be labelled by the partitions of $q-1$. If j is a partition of $q-1$ labelling the conjugacy class of β then $\delta(\beta,1)=j_1+1$ and $\delta(\beta,i)=j_i$ for $i=2,\ldots,q-1$ for all $\beta\in S_{[1]}S_{[q]\setminus[1]}$. This yields that $c_i(\beta)=1+\sum_{d\mid i}dj_d$. Summing over conjugacy classes in Equation (2.10) gives:

$$N(p,q)=\sum_{j\vdash q-1}\left(\prod_{i=1}^{q-1}j_i!\,i^{j_i}\right)^{-1}\mathcal{Z}\left(S^{\times 2}_{[p]\setminus[q]};\left(1+\sum_{d\mid 1}dj_d\right),\ldots,\left(1+\sum_{d\mid(p-q)^2}dj_d\right)\right),\tag{2.11}$$

where it was used again that the size of each conjugacy class is

$$\frac{(q-1)!}{\prod_{i=1}^{q-1}j_i!\,i^{j_i}}.$$

According to Equation (2.3) the cycle index of $S^{\times 2}_{[p]\setminus[q]}$ is

$$\mathcal{Z}(S^{\times 2}_{[p]\setminus[q]})=\sum_{k\vdash(p-q)}\left(\prod_{i=1}^{p-q}k_i!\,i^{k_i}\right)^{-1}\prod_{a,b=1}^{p-q}x_{\operatorname{lcm}(a,b)}^{k_ak_b\gcd(a,b)}.\tag{2.12}$$

Substituting (2.12) into (2.11) gives the formula for $N(p,q)$. □

The previous lemma provides the essential information to determine a formula for the number of 3-nilpotent semigroups of order n up to isomorphism.

Theorem 2.3.5 *Let $n \in \mathbb{N}$ and $N(p,q)$ be defined as in Lemma 2.3.4, that is*

$$N(p,q) = \sum_{j \vdash q-1} \sum_{k \vdash p-q} \left(\prod_{i=1}^{q-1} j_i!\, i^{j_i} \prod_{i=1}^{p-q} k_i!\, i^{k_i} \right)^{-1} \prod_{a,b=1}^{p-q} \left(1 + \sum_{d \mid \operatorname{lcm}(a,b)} d j_d \right)^{k_a k_b \gcd(a,b)} .$$

Then the number of non-isomorphic 3-nilpotent semigroups with n elements equals

$$|\widehat{Z_n}| = \sum_{m=2}^{m_0(n)} \left(N(n,m) - N(n-1,m-1) \right),$$

where

$$m_0(n) = \left\lfloor n + 1/2 - \sqrt{n - 3/4} \right\rfloor .$$

Proof: The set of orbits counted by $N(p,q)$ includes those with functions which do not take every element in $[q] \setminus [1]$ as image. The number of these orbits equals $N(p-1,q-1)$, the number of orbits of functions with one fewer element in the image set. According to Lemma 2.3.3 the number of non-isomorphic semigroups in $Z_n(m)$ equals the number of orbits of functions having $[m] \setminus [1]$ in their image set. Both are $N(n,m) - N(n-1,m-1)$ due to the argument above. Summing over m as in (2.9) proves the theorem. $\square$

In the next part, the considerations to determine the number of 3-nilpotent semigroups up to isomorphism are repeated up to equivalence.

Lemma 2.3.6 *Two semigroups $S(\psi,1)$ and $S(\chi,1)$ in $Z_n(m)$ are equivalent if and only if ψ and χ lie in the same orbit under the action of the power group $(2S_{[n]\setminus[m]}^{\times 2})^{S_{[1]}S_{[m]\setminus[1]}}$.*

Proof: ($\Rightarrow$) If $S(\psi)$ and $S(\chi)$ are isomorphic use Lemma 2.3.3. So assume that $\pi \in S_n$ is an anti-isomorphism between the two semigroups. As in the proof of Lemma 2.3.3 let σ and τ denote the permutations on $[m]$ and $[n] \setminus [m]$ induced by π. Then, for all $x, y \in [n] \setminus [m]$,

$$\psi(x,y) = (\psi(x,y)^{\pi})^{\pi^{-1}} = (\chi(y^{\pi}, x^{\pi}))^{\pi^{-1}} = (\chi(y^{\sigma}, x^{\sigma}))^{\tau^{-1}} = \chi^{(\sigma;\tau^{-1})}(y,x).$$

Hence $\psi = \chi^{(\bar{\sigma};\tau^{-1})}$.

($\Leftarrow$) If $\psi = \chi^{(\sigma;\tau)}$ with $\sigma \in S_n$, then $S(\psi)$ and $S(\chi)$ are isomorphic semigroups by Lemma 2.3.3. So assume that $\psi = \chi^{(\bar{\sigma};\tau)}$ with $\sigma \in S_n$. Choose $\pi \in S_n$ depending on σ and τ as in the proof of Lemma 2.3.3. Since

$$\psi(x, y) = \chi^{(\bar{\sigma};\tau)}(x, y) = \chi^{(\sigma;\tau)}(y, x),$$

the bijection π is an anti-isomorphism from $S(\chi)$ to $S(\psi)$. $\square$

The previous lemma allows one to determine a formula for the number of 3-nilpotent semigroups up to equivalence by finding the number of non-equivalent functions under the power group action.

Theorem 2.3.7 *Let $n \in \mathbb{N}$ and define $L(p, q)$ for $1 \leq q < p \leq n$ to be*

$$L(p, q) = \frac{1}{(q-1)!} \sum_{j \vdash (q-1)} \mathcal{Z}\left(2S^{\times 2}_{[p]\setminus[q]}; 1 + \sum_{d|1} dj_d, 1 + \sum_{d|2} dj_d, \ldots, 1 + \sum_{d|(p-q)^2} dj_d\right).$$

Then the number of non-equivalent 3-nilpotent semigroups with n elements equals

$$|\overline{Z_n}| = \sum_{m=2}^{n-1} (L(n, m) - L(n-1, m-1)).$$

Proof: The proof follows the exact same steps as the ones of Lemma 2.3.4 and Theorem 2.3.5, simply replacing the cycle index of $S^{\times 2}_{[p]\setminus[q]}$ with the cycle index of $2S^{\times 2}_{[p]\setminus[q]}$ given in (2.4). $\square$

When knowing the number of semigroups in a class (or even other mathematical objects) up to isomorphism and up to equivalence, one easily obtains the number of self-dual semigroups in that class. It was shown in the proof of Lemma 1.1.3 that, if l is the number up to equivalence, then $2l$ is the number of non-isomorphic semigroups counting self-dual semigroups twice. Thus the number of 3-nilpotent, self-dual semigroups of order n up to isomorphism equals $2|\overline{Z_n}| - |\widehat{Z_n}|$. Remember that for self-dual semigroups it is unnecessary to distinguish between 'non-isomorphic' and 'non-equivalent', since each anti-isomorphism is at the same time an isomorphism and vice versa.

An important subclass of (self-dual) semigroups are commutative semigroups. To obtain the number of 3-nilpotent, commutative semigroups with n elements, the analysis from above is repeated on a different set of functions. Define

$$CZ_n = \{S \in Z_n \mid xy = yx \text{ for all } x, y \in S\}$$

and analogously $CZ_n(m)$ as subset of $Z_n(m)$. First, it is straightforward to count the semigroups in CZ_n in the same way as it was done for Z_n in Lemma 2.3.1.

Lemma 2.3.8 *For $n \in \mathbb{N}$ the number of different commutative 3-nilpotent semigroups on $[n]$ equals*

$$|CZ_n| = \sum_{m=2}^{m_0^{(c)}(n)} \binom{n}{m} m \sum_{i=0}^{m-1} (-1)^i \binom{m-1}{i} (m-i)^{(n-m)(n-m+1)/2},$$

where

$$m_0^{(c)}(n) = \left\lfloor n + 3/2 - \sqrt{2n + 1/4} \right\rfloor .$$

Proof: The proof follows the same steps as the one of Lemma 2.3.1. For a commutative semigroup $S(\psi, z)$ the function $\psi : ([n] \setminus B)^2 \to B$ is defined by its values on pairs (i, j) with $i \leq j$. If m denotes $|B|$ there are $(n-m)(n-m+1)/2$ such pairs.

For a fixed set B only functions taking all elements in $B \setminus \{z\}$ as value are counted, which implies the condition $m - 1 \leq (n-m)(n-m+1)/2$. Reformulating this inequality yields the parameter $m_0^{(c)}(n)$. $\square$

Again, it is claimed in [JMS91, Theorem 15.8], the formula in the previous lemma is for a subclass of CZ_n (see the comment after Lemma 2.3.1). In addition, the parameter $m_0^{(c)}(n)$ is corrected.

For each $S(\psi, 1) \in CZ_n(m)$ the equality $\psi(i, j) = \psi(j, i)$ holds for all $i, j \in [n] \setminus [m]$. Hence, the induced function ψ' on $([n] \setminus [m])^{\{2\}}$ for which $\psi'\{i\} = \psi(i, i)$ and $\psi'\{i, j\} = \psi(i, j)$ for $i \neq j$ is well-defined. Each function from $([n] \setminus [m])^{\{2\}}$ to $[m]$ is induced by some $\psi : ([n] \setminus [m])^2 \to [m]$.

Lemma 2.3.9 *Two semigroups $S(\psi, 1)$ and $S(\chi, 1)$ in $CZ_n(m)$ are isomorphic if and only if ψ' and χ' lie in the same orbit under the action of the power group $(S_{[n]\setminus[m]}^{\{2\}})^{S_{[1]} S_{[m]\setminus[1]}}$.*

Proof: Using Lemma 2.3.3 it suffices to notice that ψ' and χ' are in the same orbit under $(S^{\{2\}}_{[n]\setminus[m]})^{S_{[1]}S_{[m]\setminus[1]}}$ if and only if ψ and χ are in the same orbit under $(S^{\times 2}_{[n]\setminus[m]})^{S_{[1]}S_{[m]\setminus[1]}}$. $\square$

The result on the number of commutative 3-nilpotent semigroups follows in the obvious way.

Theorem 2.3.10 *Let $n \in \mathbb{N}$ and define $K(p,q)$ for $1 \leq q < p \leq n$ to equal*

$$\sum_{\substack{j \vdash q-1 \\ k \vdash p-q}} \left(\prod_{i=1}^{q-1} j_i!\, i^{j_i} \prod_{i=1}^{p-q} k_i!\, i^{k_i} \right)^{-1} \prod_{a=1}^{\lfloor \frac{n}{2} \rfloor} c(a)^{k_{2a}} c(2a)^{ak_{2a}} \prod_{a=1}^{\lfloor \frac{n+1}{2} \rfloor} c(2a-1)^{ak_{2a-1}} \prod_{a<b} c(l_{a,b})^{k_a k_b g_{a,b}}$$

with $l_{a,b} = \mathrm{lcm}(a,b)$, $g_{a,b} = \gcd(a,b)$, and $c(x) = 1 + \sum_{d|x} dj_d$. Then the number of non-equivalent 3-nilpotent, commutative semigroups with n elements equals

$$|\overline{CZ_n}| = \sum_{m=2}^{n-1} (K(n,m) - K(n-1,m-1)).$$

Proof: The proof follows the same steps as the one of Lemma 2.3.4 and Theorem 2.3.5, only replacing the cycle index of $S^{\times 2}_{[p]\setminus[q]}$ with the cycle index of $S^{\{2\}}_{[p]\setminus[q]}$ given in (2.5). $\square$

Tables with the numbers of 3-nilpotent semigroups on $[n]$ in all variations occurring in this section are compiled in Appendix A.1 for small values of n. They were computed using the function `Nr3NilpotentSemigroups` from Smallsemi [DM10] (see Section 4.2).

What hope is there to extend the methods in this section to count 4-nilpotent semigroups? Harrison [Har66] applied the enumeration techniques presented in Section 2.3 to sets with an arbitrary (finite) number of operations each of finite degree, known as universal algebras. He mentions that in the next step towards formulae for the number of other algebraic structures – such as semigroups, groups and rings – associativity has to be studied first. It may seem like a step towards this aim was made in this section. Note first, that the definition of nilpotency rank is applicable to magmas in general. Then one possible point of view is, that the presented methods were used to count 3-nilpotent magmas, which happened to be semigroups. To extend the approach to 4-nilpotent magmas would work, but this

Table 2.1 Ratio of lower bound and actual number of 3-nilpotent semigroups

n	3	4	5	6	7	8	9	10
$\frac{\lvert Z_n \rvert}{2n!\lvert\overline{Z_n}\rvert}$	0.5	0.46875	0.58135	0.80651	0.96703	0.99757	0.99974	0.99996
$\frac{\lvert CZ_n \rvert}{n!\lvert\overline{CZ_n}\rvert}$	1	0.7	0.58696	0.60188	0.72124	0.85657	0.93649	0.97449

would not give the numbers of 4-nilpotent semigroups. Already counting different semigroups on $[n]$ of any particular type is in general difficult due to associativity. Without a set of functions to apply Theorem 2.2.7 to, the presented methods are not applicable.

With $|\overline{Z_n}|$ a new (presumably tight) lower bound for the number of non-equivalent semigroups on $[n]$ is provided. The enumeration formula in Theorem 2.3.7 yields not only the total number of 3-nilpotent semigroups, but contains as well the numbers of semigroups with any specified bijection as automorphism or anti-automorphism. Studying this information in detail it should be possible to answer the following question,

Question 2.3.11 Do asymptotically all non-equivalent, 3-nilpotent semigroups on $[n]$ have trivial automorphism group?

or at least the weaker version:

Question 2.3.12 Do asymptotically all different, 3-nilpotent semigroups on $[n]$ have trivial automorphism group?

These questions relate to the observation that the former bound $|Z_n|/2n!$ for the number of semigroups seems to converge rapidly towards $|\overline{Z_n}|$ as illustrated in Table 2.1. With the formulae for both $|Z_n|$ and $|\overline{Z_n}|$ available, the answer to the questions is within reach. An important result for asymptotic behaviour of 3-nilpotent semigroups is given in [KRS76]. When n tends to infinity, the proportion of semigroups in Z_n with $n - n/(2\ln n)$ generators tends to 1. Further research around this topic might ultimately lead to a proof that asymptotically all non-equivalent semigroups on $[n]$ have trivial automorphism group. Such results are known for many other combinatorial and algebraic structures, for example for graphs [ER63].

The results in this section will become important again in the enumeration of all semigroups up to equivalence. A comparison between the numbers for 3-nilpotent semigroups and all semigroups is done later.

3 Diagonals

In Chapter 1 representation of a binary operation via its multiplication table was introduced. Equivalence of multiplication tables was defined based on isomorphisms and anti-isomorphisms between the represented algebraic structures, and expressed via an action of $S_n \times C_2$ on the set of tables representing magmas with n elements. Under this action each entry on the diagonal of a table is mapped to an entry on the diagonal of the image. Thus the action on tables induces an action on diagonals of tables. Plemmons used this independence of the diagonal from the rest of the table in the search for semigroups of order 6 [Ple67]. The idea is to consider only diagonals that are not equivalent under the induced action, and then to perform a separate search for each diagonal. The idea was adopted by Satoh, Yama, and Tokizawa for the search of semigroups of order 8 [SYT94]. In their backtrack algorithm diagonal positions of the table are considered first. They state that they have determined '660 representatives of the diagonal positions'. No further information on the diagonals is given in either of the references [Ple67, SYT94].

For Plemmons, as well as Satoh et al., the search for non-equivalent diagonals was one step in the process of finding multiplication tables of semigroups. The numbers of diagonals were known before, and, due to the small orders, this step was comparatively simple. While one purpose of this chapter is to prepare for this step in the search for semigroups, the content goes beyond this. Partially, diagonals are studied independently, but also a better insight into the role the diagonal plays in the multiplication of a semigroup is gained.

In the first section of this chapter a way to find non-equivalent diagonals using their underlying structure, rather than a mere search, is presented. The correspondence of diagonals to a certain class of directed graphs is employed to develop an algorithm giving one diagonal of every equivalence class. Slight changes to the algorithm let it output another set of diagonals, being non-equivalent with respect to a different group. Results from implementations of the original algorithm and its adaptation into GAP [GAP08] are provided.

In the second section diagonals are studied in connection with multiplication tables. The starting point is the situation which will be the most common initial setup in the search for semigroups: given is a multiplication table in which only the diagonal positions are known. The first question at this point is whether a diagonal appears at all in the multiplication table of a semigroup; and if it does, how it influences the structure of the semigroup. Criteria answering the first question form the majority of the second section.

3.1 Constructing Diagonals

In the search for multiplication tables of semigroups of order 6, Plemmons introduced the idea of choosing the entries on the diagonal before the search [Ple67]. Subsequently, he performed a separate search for each choice of diagonal. Even without going into the details of the search, it is obvious that not all n^n different diagonals have to be considered, if one is interested in the semigroups of order n up to equivalence. In this section an algorithm giving a smaller set of diagonals is explained. More precisely, a correspondence between diagonals and a certain class of directed graphs will be established together with a notion of equivalence for diagonals. It follows a closer look at the type of directed graphs involved in the correspondence. The insight is then utilised in an algorithm that outputs a list of all diagonals up to equivalence. The section finishes with a variation of the algorithm adjusting it to a related problem.

The diagonal of a table $T = (T_{i,j})_{1\leq i,j\leq n}$ is the n-tuple $(T_{1,1}, T_{2,2}, \dots, T_{n,n})$. Having values in $[n]$ the diagonal corresponds to the function

$$f_T : [n] \to [n], i \mapsto T_{i,i}.$$

For convenience this corresponding function shall be used in the following instead of the diagonal itself. As every function $f : [n] \to [n]$ is uniquely defined by the n-tuple $(f(1), f(2), \dots, f(n))$ the correspondence is indeed a bijection. The action of $S_n \times C_2$ on multiplication tables as defined by (1.1) and (1.2) induces an action on diagonals, since every diagonal entry is mapped to a diagonal entry. For a function f_T and an element $g \in S_n \times C_2$ the induced action of g on f_T is given by f_{T^g} and denoted by f_T^g. Note that transposing a table does not influence the diagonal, making it superfluous to consider both isomorphisms and anti-isomorphisms.[1] For a permutation $\pi \in S_n$ both (π, e) and (π, c) – where e denotes the trivial and c the non-trivial element in C_2 – have the same effect on all diagonals. Hence, S_n-equivalence of diagonals and $(S_n \times C_2)$-equivalence yield the same orbits. In the following it suffices to consider the induced action of elements in S_n on diagonals.

A digraph Γ_T can be associated with a diagonal of the table T. The vertex set of Γ_T is $[n]$ and the edge set is $\{(i, f_T(i)) \mid i \in [n]\}$. The correspondence can be used to connect equivalence of diagonals to isomorphism of graphs.

Lemma 3.1.1 *Let S and T be the multiplication tables of two binary operations on $[n]$. Then Γ_S is isomorphic to Γ_T if and only if f_S is S_n-equivalent to f_T.*

Proof: $(\Rightarrow)$: There exists $\pi \in S_n$ sending the edge set of Γ_T to the edge set of Γ_S, that is $\{(i^\pi, f_T(i)^\pi) \mid i \in [n]\} = \{(i, f_S(i)) \mid i \in [n]\}$. The set on the left hand side can be rewritten as $\{(i, f_T(i^{\pi^{-1}})^\pi) \mid i \in [n]\}$. Using $f_T(k) = T_{k,k}$ it follows that

$$f_T(i^{\pi^{-1}})^\pi = (T_{i^{\pi^{-1}}, i^{\pi^{-1}}})^\pi = (T^\pi)_{i,i}.$$

Hence the edge set also equals $\{(i, f_{T^\pi}(i)) \mid i \in [n]\}$ and a simple comparison of functions yields that $f_{T^\pi} = f_S$.

$(\Leftarrow)$: There exists $\pi \in S_n$ such that $f_{T^\pi} = f_S$. Going backwards through the arguments of the proof for the other direction it follows π is an isomorphism from Γ_T to Γ_S. □

The next aim is to obtain a set of representatives of the equivalence classes of diagonals. To do this, a set of non-isomorphic digraphs representing functions shall be constructed. Note that the number of non-equivalent functions on $[n]$ are long known. A formula in the style of the results in Section 2.3 is given in [Dav53, Theorem 6] together with the terms for n up to 5. Adjusted to the notion used in this thesis the formula reads

$$\sum_{j \vdash n} \left(\prod_{i=1}^{n} j_i! i^{j_i} \right)^{-1} \prod_{a=1}^{n} \left(\sum_{d \mid a} d j_d \right)^{j_a} .$$

[1] In other words, the action of $S_n \times C_2$ on diagonals is not faithful.

More terms together with a counting series are given in [HP73, Table 3.4.1]. In the same reference one can find as well numbers of *functional digraphs*,[2] corresponding to functions without fixed point. These are of interest as every finite semigroup contains an idempotent (Remark 1.2.3), and therefore functional digraphs lead to diagonals that do not appear in the multiplication table of any semigroup. Numbers of non-equivalent functions with fixed point are presented in Table 3.1. They are simply calculated as the difference between the numbers of digraphs representing functions and the numbers of functional digraphs. For $n = 8$ one finds that there are 660 non-equivalent functions with fixed point, which apparently correspond to the 660 diagonals determined by Satoh et al. in [SYT94].

Table 3.1 Numbers of non-equivalent functions from $[n]$ to $[n]$

n	1	2	3	4	5	6	7	8	9	10
functions	1	3	7	19	47	130	343	951	2 615	7 318
–, with fixed point	1	2	5	13	34	90	243	660	1 818	5 045
functional digraphs	0	1	2	6	13	40	100	291	797	2 273

To actually obtain a set of non-equivalent functions using the correspondence to digraphs, it would be easy to start with all functions to get all digraphs. Only this is not helpful, as deciding graph isomorphism is, in general, a hard problem. On the other hand, two labelled graphs are isomorphic if and only if their corresponding unlabelled graphs are identical. For the special class of digraphs representing functions the types of structures involved can be described in a way that allows an inductive construction of non-isomorphic graphs. To make the description easier, the term 'rooted tree' – usually referring to an undirected tree with a distinguished vertex – is defined for digraphs as follows.

Definition 3.1.2 A digraph is a *rooted tree*, if it is connected, there exists a unique vertex, called the *root*, with outdegree 0, and all other vertices have outdegree 1.

A connected graph with n vertices and $n-1$ edges is a tree. From Definition 3.1.2 it is clear that all edges point in the direction of the root, in the sense that the end vertex of every edge lies on the path from the start vertex to the root. In

[2]The term 'functional digraph' is not used consistently throughout the literature. It sometimes refers to all digraphs representing a function.

the literature it is more common that the edges in a rooted tree are pointing away from the root. In any case a characterisation of rooted trees is well known.

Lemma 3.1.3 *Rooted trees on n vertices are in one-one correspondence with forests of rooted trees on $n-1$ vertices. If Γ is a rooted tree with root r, then $\Gamma \setminus \{r\}$ is the corresponding forest of rooted trees.*

The previous lemma is illustrated in Figure 3.1. Now, a well known description of digraphs corresponding to functions can be given.

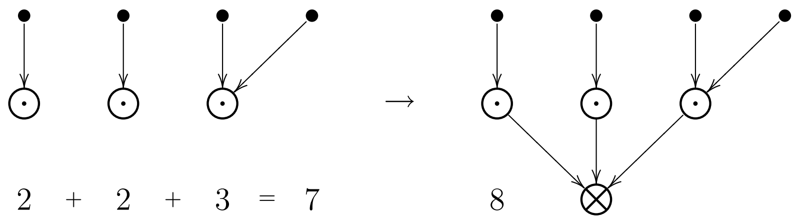

Figure 3.1 Building rooted trees
The four rooted trees on the left with a total of seven vertices are assembled to one rooted tree with eight vertices by introducing a new root vertex.

Lemma 3.1.4 *Let Γ be a digraph representing a function from $[n]$ to $[n]$. Then every connected component of Γ contains exactly one cycle. After removing the edges in the cycles, the connected components are rooted trees with the vertices from the cycle as roots.*

Proof: As Γ represents a function every vertex has outdegree 1. This fact is used repeatedly throughout this proof.

Let C be a connected component of Γ and consider an infinite walk in C. An infinite walk in Γ is determined by the first vertex in the walk (because every vertex in the walk has outdegree 1). Moreover, as Γ is finite there has to be repetition in the walk, showing that there is a cycle in C. To prove uniqueness consider a sequence of vertices $(v_1, v_2, \dots, v_r)$ without repetition such that v_1 and v_r are both in cycles of C and for all $1 \le i \le r-1$ either (v_i, v_{i+1}) or (v_{i+1}, v_i) is an edge. The sequence is a path in the underlying graph of Γ. Such a sequence must exist for every two vertices in C due to connectedness. Assume v_1 and v_r are from different cycles. Then there is a minimal k with v_k not in the same cycle as v_1. Thus (v_{k-1}, v_k) is not an edge but (v_k, v_{k-1}) is. Then (v_k, v_{k+1}) is not an edge as v_k has outdegree 1 and $v_{k-1} \neq v_{k+1}$. Inductively (v_{r-1}, v_r) is not an edge. Thus (v_i, v_{i-1}) is an edge for all $k \leq i \leq r$ making v_{k-1} a vertex in a cycle with both v_1 and v_r, a contradiction.

Removing the edges from the unique cycle in C, the connected components do not have a cycle and are therefore trees. Let u be a vertex from the cycle. As the unique edge starting at u has been removed, its outdegree is 0, while the outdegree of all other vertices is still 1. Hence, the connected component is a rooted tree with u as root. □

Every graph isomorphism maps connected components to connected components. Thus to obtain all graphs up to isomorphism, it suffices to know the connected components up to isomorphism. It is easy to see how the components look in the case of digraphs representing a function. As the knowledge will be needed later, it is presented in the following lemma.

Lemma 3.1.5 *Let Γ and Δ be two connected digraphs representing functions from $[n]$ to $[n]$. Denote the length of the cycle of Γ by r and let $(v_1, v_2, \dots, v_r)$ be a path through the cycle. Likewise, denote the length of the cycle of Δ by s and let $(u_1, u_2, \dots, u_s)$ be a path through its cycle. Moreover, let $(R(v_1), R(v_2), \dots, R(v_r))$ and $(S(u_1), S(u_2), \dots, S(u_s))$ be the rooted trees connected to the cycles, $R(v_i)$ with root v_i and $S(u_i)$ with root u_i.*

Then Γ and Δ are isomorphic if and only if $r = s$ and there is a power π of the permutation $(1\,2 \cdots r)$ such that $R(v_{i^\pi}) \cong S(u_i)$ for all $1 \leq i \leq r$.

Proof: $(\Rightarrow)$: It is clear that the cycle of Γ has to be mapped to the cycle of Δ by every isomorphism σ and thus $r = s$. Moreover if $v_1^\sigma = u_j$ then $v_i^\sigma = u_{i^\pi}$ for $\pi = (1\,2 \cdots r)^{(j-1)}$ and thus $R(v_i) \cong S(u_{i^\pi})$.

$(\Leftarrow)$: Let σ_i be an isomorphism from $R(v_{i^\pi})$ to $S(u_i)$ for all $1 \leq i \leq r$. Define σ to map the vertex v from Γ to $\sigma_i(v)$ where σ_i is the unique isomorphism with v in its domain. Then σ is a bijection from the vertex set of Γ to the vertex set of Δ preserving not only the edges in the trees but as well the edges in the cycle. Thus σ is an isomorphism. □

From the previous lemma and Lemma 3.1.4 it is known what connected digraphs representing functions look like and when they are isomorphic. An algo-

Algorithm 1

Construct the connected digraphs with N vertices and a cycle of length K

Require: $K \leq N$

$L \leftarrow [\,]$ {initialise output as empty list}
$C \leftarrow$ cycle of length K {vertices labelled 1 to K}
for all $P \in \textsc{Partitions}(N, K)$ **do** {the partition specifies the sizes of rooted trees at the vertices of the cycle}
    for all $(T_1, T_2, \ldots, T_K) \in \textsc{Forests}(P)$ **do**
        $\mathcal{F} \leftarrow \{(T_{1^\pi}, T_{2^\pi}, \ldots, T_{K^\pi}) \mid \pi \in S_K\}$ {set of all arrangements of $(T_1, T_2, \ldots, T_K)$}
        for all $\mathcal{O} \in \textsc{Orbits}(\langle (1\,2\,\cdots\,K)\rangle, \mathcal{F})$ **do**
            $(R_1, R_2, \ldots, R_K) \leftarrow$ representative of $\mathcal{O}$ {arbitrary element in the orbit}
            $D \leftarrow$ copy of C
            for all $i \in \{1, 2, \ldots, K\}$ **do**
                $D \leftarrow D$ merged with R_i by identifying vertex i with the root of R_i
            end for
            add D to list L
        end for
    end for
end for
return L

rithm constructing one such digraph of every isomorphism type is given as Algorithm 1. As prerequisite three algorithms Partitions, Forests, and Orbits are assumed to exist.

Partitions takes two positive integers N and K as input and outputs all partitions of N with K summands of positive integers.

Forests takes a partition $a_1 + a_2 + \cdots + a_N$ as input and outputs all forests up to isomorphism consisting of N trees $T_1, T_2, \ldots, T_N$, where T_i has a_i vertices for $1 \leq i \leq N$.

Orbits takes a group and a set as input; the set being closed under the action of the group. It outputs the orbits on the set under the action of the group.

Lemma 3.1.6 *Algorithm 1 is correct.*

Proof: According to Lemma 3.1.5 for two connected components to be isomorphic they need to have the same forest attached to the vertices in the cycle. Thus the digraphs constructed inside the loop starting at line 4 can only be isomorphic to a digraph constructed in the same run. On the other hand, for every connected digraph there is a run through this loop with the same forest as given by the digraph. It remains to be shown that in lines 7 to 12 one representative for every type of connected digraph with $(T_1, T_2, \ldots, T_K)$ as forest is constructed. The set $\mathcal{F}$ defined in line 5 represents all connected digraphs with $(T_1, T_2, \ldots, T_K)$ as forest. The orbit calculation in line 6 follows the result from Lemma 3.1.5, putting two ordered tuples in the same orbit if and only if they represent isomorphic digraphs. Thus taking one representative in line 7 and constructing the corresponding digraph in lines 8 to 11 leads to the required set. □

Compared with the brute force approach of calculating the orbits of the set of all functions from $[n]$ to $[n]$, the orbit calculations in Algorithm 1 are very easy, making this algorithm a big improvement. Nonetheless a simple change can be made to improve its performance even more. The idea is that for the majority of forests it is clear how the orbits in line 6 look. Thus the orbit calculation – still the bottleneck of the algorithm – can be avoided. In the simplest case $(T_1, T_2, \ldots, T_K)$ contains some isomorphism type of tree, T, only once. Then each orbit of $\mathcal{F}$ (from line 5 of Algorithm 1) under $\langle(1\,2 \cdots K)\rangle$ contains exactly one tuple that has T in first position. Thus, of all ordered tuples arising from $(T_1, T_2, \ldots, T_K)$ the ones starting with T form a set of representatives.

An implementation based on Algorithm 1 in GAP was used to obtain all diagonals with a fixed point up to equivalence of size n for $1 \leq n \leq 18$. Table 3.1 contains their numbers up to $n = 10$. The implementation uses existing functionality for Partitions and Orbits, while Forests was implemented anew, inductively constructing trees as illustrated in Figure 3.1. The orbit calculation in line 6 is avoided in the case where some type of tree appears only once in $(T_1, T_2, \ldots, T_K)$. (Trying to use a generalisation of this idea, when the number of times some isomorphism type T of tree appears in $(T_1, T_2, \ldots, T_K)$ and the length of the cycle K are coprime, makes the implementation of the algorithm complicated and thus prone to errors, while being outperformed by the highly developed orbit calculation available in GAP.) The code is available on the attached DVD, see Appendix C.

An interesting variation of the original problem is to consider partial functions from $[n-1]$ to $[n-1]$. As before, each such function defines a digraph, whose edges are given by the partial function. The outdegree for each vertex is then at most 1. As on the other hand a digraph, in which each vertex has outdegree at most 1, defines a partial function, the connection is again a correspondence.

Obviously, every function can be considered as a partial function, and, making arbitrary choices for undefined values, a partial function can be completed to a function. The latter means for the corresponding digraphs that the graph of a partial function is a subgraph of the graph of some function. This observation makes the analogue of Lemma 3.1.4 in the context of partial functions a corollary.

Corollary 3.1.7 *Let Γ be a digraph representing a partial function from $[n]$ to $[n]$. Each connected component of Γ is either a rooted tree or contains exactly one cycle. After removing the edges in the cycles, the connected components are rooted trees and each vertex from a cycle is a root.*

Proof: Consider which consequences removing edges from a digraph representing a function can have. Removing an edge from a cycle leaves the component connected, but with one fewer edge than vertices it becomes a tree; removing an edge from outside a cycle creates a separate component that is a tree. If in a subsequent step an edge is removed from a tree, it splits into two trees. Now the statement follows from Lemma 3.1.4. □

Trees are the only new type of connected components appearing in the previous lemma in comparison with Lemma 3.1.4. They are known as prerequisite for Algorithm 1, inductively constructed as illustrated in Figure 3.1. Again, a program that creates the connected components and assembles them in all non-isomorphic ways was implemented in GAP. This program was used to obtain non-equivalent partial functions up to order 18. Table 3.2 contains the numbers up to 11. More terms are available at the On-Line Encyclopedia of Integer Sequences [Slo09].

Not only can one construct partial functions in the presented way, but there is also a connection to diagonals of multiplication tables. To explain this in detail, it would be necessary to anticipate the search for semigroups. Instead, a rough idea is given. Extending a partial function on $[n-1]$ to a function on $[n]$ by mapping each element with undefined image to n leads to a diagonal with a fixed point, namely n. Under this construction certain pairs of non-S_{n-1}-equivalent partial

Table 3.2 Numbers of non-equivalent partial functions from $[n]$ to $[n]$

n	1	2	3	4	5	6	7	8	9	10	11
partial functions on $[n-1]$	1	2	6	16	45	121	338	929	2 598	7 261	20 453

functions result in S_n-equivalent diagonals. Equivalence of diagonals was defined using the action of S_n induced from the action on tables. If a semigroup has n as distinguished idempotent – an identity or a zero element say – one has to use the action of S_{n-1} instead. Under this subgroup the construction from partial functions becomes the right choice to obtain a set of non-equivalent diagonals.

The final remark in this section concerns an alternative approach to the construction of non-isomorphic graphs. In [Rea78] Read introduced the idea of what he called an ‘orderly algorithm’. The main idea is to introduce a linear ordering on the graphs to obtain a notion of canonicity. In addition every canonical graph has to arise from a smaller canonical one by adding vertices and edges. It is an inductive process which includes some form of isomorphism testing to verify canonicity. While the concept has proven useful for many different classes of graphs – and for other structures as demonstrated for example in [HR02] – it seems that for digraphs representing functions the presented approach is superior. Due to the special structure described in Lemma 3.1.4 an inductive process is only needed for building rooted trees as a prerequisite for Algorithm 1.

3.2 Analysing Diagonals

This section is dominated by the question as to whether a given diagonal appears in a table defining an associative multiplication. Of course this question is decidable, as it can, in theory, be answered by testing all finitely many ways of completing the table. Such a method is impractical even for small tables, because of the vast number of tests necessary. In this section more practical criteria using only the diagonal itself are presented. These criteria do not apply to every diagonal. Using them leaves the above question undecided for some cases.

After giving experimental data and introducing the concept of partial multiplication, there are two subsections addressing the initial question from two directions. In Subsection 3.2.1, diagonals that do not appear in any associative multiplication table are identified. Subsection 3.2.2 contains ways to build diagonals, that do appear, from smaller diagonals.

More generally, the connection between the diagonal of a multiplication table and the semigroup defined by the table is studied. Some connections are very obvious, like between the diagonal and the number of idempotents in the semigroup. When considering diagonals as functions, idempotents correspond to the fixed points, and it was mentioned before that every finite semigroup has at least one idempotent (Remark 1.2.3). The connection that will be exploited is between the diagonal and the monogenic subsemigroups of the semigroup. In first instance, for every element x only powers of the form x^{2^i} are computable from the diagonal. This indicates that two semigroups with structurally different sets of monogenic subsemigroups might still have identical diagonals. In particular, to draw conclusions about Green's equivalences of a semigroup just by knowing the squares of all elements is essentially impossible. For example, the two monogenic semigroups of order 2 – the cyclic group and the zero semigroup – both have a constant function as diagonal. The two elements in the cyclic group form an $\mathcal{H}$-class, while the two elements in the zero semigroup are not even $\mathcal{D}$-related. One might argue that the situation is different when there exists a unique semigroup with a given diagonal. An example of such a case is given in Figure 3.2.

$$\begin{pmatrix} 1 & 1 & 1 & 4 & 5 & 5 & 5 & 4 & 5 \\ 1 & 1 & 1 & 4 & 5 & 5 & 5 & 4 & 6 \\ 1 & 1 & 2 & 4 & 5 & 5 & 6 & 4 & 7 \\ 4 & 4 & 4 & 5 & 1 & 1 & 1 & 5 & 1 \\ 5 & 5 & 5 & 1 & 4 & 4 & 4 & 1 & 4 \\ 5 & 5 & 5 & 1 & 4 & 4 & 4 & 1 & 4 \\ 5 & 5 & 6 & 1 & 4 & 4 & 4 & 1 & 4 \\ 4 & 4 & 4 & 5 & 1 & 1 & 1 & 6 & 2 \\ 5 & 6 & 7 & 1 & 4 & 4 & 4 & 2 & 8 \end{pmatrix}$$

Figure 3.2 Unique associative multipl. table with diagonal $(1, 1, 2, 5, 4, 4, 4, 6, 8)$

As for most of the research contained in this thesis, experimental results were one of the starting points in the study of the influence of the diagonal on the other entries in a multiplication table of a semigroup. The data collected in Table 3.3 shows that, for the inspected orders, many diagonals do not allow the remaining

Table 3.3 Numbers of diagonals appearing in associative multiplication tables

n	1	2	3	4	5	6	7	8	9	10
diagonals	1	3	7	19	47	130	343	951	2 615	7 318
–, of semigroups	1	2	5	11	26	60	138	319	740	1 720

positions in a multiplication table to be filled such that an associative multiplication is defined.[3] Interestingly, the sets of diagonals do not reduce further if one allows only commutative semigroups.

The situation where only the diagonal of a multiplication table is given – or in other words only the squares of elements are known – is described in a more general setting as follows. A *partial multiplication* is a binary operation that is not necessarily defined for every pair of elements. Each multiplication table with entries given for some positions, defines a partial multiplication. The product of a pair of elements is known whenever the respective entry is available. In such a setting the usual definition of associativity has to be generalised, as not every expression will evaluate.

Definition 3.2.1 Let P be a set with a partial multiplication defined on it, and let v and w be two words in the elements of P.

(i) The word v *contracts* to w if there exist (possibly empty) words s and r such that $v = sp_ip_jr$ and $w = sp_kr$ where $p_ip_j = p_k$ in P.

(ii) The two words v and w are *associated* if there exists a sequence of words $v = v_1, v_2, \ldots, v_{l+1} = w$ such that for all $1 \le i \le l$ either v_i contracts to v_{i+1} or v_{i+1} contracts to v_i.

The process described in Definition 3.2.1(i) will be referred to as *contraction.* The terms '*to expand*' and '*expansion*' will be used to describe the reverse process.

Definition 3.2.2 A partial multiplication is *associative* if no two distinct elements are associated.

This definition of associativity coincides with the usual one in case every product is defined. Furthermore, every partial multiplication which can be completed

[3]See Chapter 5 for details on how this data was collected.

$$\begin{pmatrix} 2 & \diamond \\ \diamond & 1 \end{pmatrix}$$

Figure 3.3 An associative partial multiplication; $\diamond$ denotes undefined products

to a multiplication of a semigroup is associative. The converse does not hold. A minimal example for an associative partial multiplication whose table cannot be completed to a multiplication table of a semigroup is given in Figure 3.3. This example can be generalised, leading back to the original question as to whether a diagonal allows an associative multiplication.

Lemma 3.2.3 *Let P be a finite set with a partial multiplication defined on it, such that exactly the squares of elements are known. Then the partial multiplication is associative.*

Proof: Let $P = \{p_1, p_2, \ldots, p_n\}$. Take two associated elements from P. Without loss of generality let p_1 be one of them and let the other be p_j for some $1 \leq j \leq n$. It shall be shown that $j = 1$. According to Definition 3.2.1(ii) there exists a sequence of words $p_1 = w_1, w_2, \ldots, w_{r+1} = p_j$ such that each word arises via contraction or expansion from its predecessor. Note that the number of contractions equals the number of expansions since w_1 and w_{r+1} have the same length.

It shall be shown by contradiction that for every expansion there is a contraction doing the opposite. Let the m-th step be the last expansion which is not undone. Thus w_m expands to w_{m+1}, meaning some p_s in w_m is replaced with $p_t p_t$, where the square of p_t equals p_s according to the partial multiplication. As $p_t p_t$ is not part of w_{r+1}, at least one p_t has to vanish at some point. This cannot happen due to any later expansions as they are all undone. On the other hand, the only possible contraction including p_t is to replace $p_t p_t$ by p_s. This would simply undo the expansion, and thus contradict the assumption. Concluding that the final word still contains $p_t p_t$ yields a contradiction. Thus all expansions are cancelled by contractions, and $p_j = p_1$, making the partial multiplication associative. $\square$

The key point in the proof for the previous lemma is that every element is a factor in at most one product of the partial multiplication. Each contraction is then uniquely defined by one element it involves. Thus the proof can be adapted to show a statement generalising Lemma 3.2.3 to all partial multiplications satisfying the above condition. Moreover, there is no need to restrict defined products to pairs of elements. One can, for example, replace 'squares of elements' in the condition of the lemma by 'k-th powers of elements'. As these situations are of no further interest in this thesis, Lemma 3.2.3 states only the restricted case.

The example in Figure 3.3 and the previous result could give the impression that Definition 3.2.2 should be revised. Even though it generalises the idea of different ways of bracketing a product for partial multiplications, it cannot be used to identify diagonals that do not appear in the multiplication of a semigroup. A quick excursion away from the aim of this section shows the usefulness of Definition 3.2.2. Instead of asking whether a table of a partial multiplication can be completed to a semigroup table, one can ask whether it embeds into the table of a semigroup. The partial table in Figure 3.3 for example, is a subtable of a Cayley table for C_3, the group with 3 elements; 1 and 2 then being the non-identity elements. Tamari used in [Tam73] that being embeddable into a semigroup is equivalent to Definition 3.2.2. Note that, even for finite partial multiplications, the embedding is often into an infinite semigroup.

Returning to the aim of this section, Lemma 3.2.3 gives a negative result for the attempt to answer which diagonals can appear in the multiplication table of a semigroup.

3.2.1 Excluded diagonals

Despite the fact that every diagonal defines an associative partial multiplication, criteria for a diagonal not to appear in the multiplication table of a semigroup can be given. Every semigroup has an idempotent and thus every function related to the diagonal of a semigroup has a fixed point (see Table 3.1).

This argument can be seen as information coming from the monogenic subsemigroups of the semigroup. Even though monogenic subsemigroups are not determined by the diagonal, some information can be deduced. In particular, certain diagonals would require too large a number of elements in a single monogenic subsemigroup and are therefore not diagonals of an associative multiplication table.

Lemma 3.2.4 *Let S be a semigroup of order n and let Γ be the digraph corre-*

sponding to the diagonal of the multiplication table of S*. If* $(v_0, v_1, \ldots, v_l)$ *is a path in* Γ *without a vertex from a cycle, then the following hold:*

(i) *the length* l *of the path is bounded by* $\lfloor \log_2(n-1) \rfloor$*;*

(ii) *if the cycle of the component containing the path is a loop, then* l *is bounded by* $\lfloor \log_2(k-1) \rfloor$*, where* k *is the number of vertices in the component.*

Proof: Consider a path $(v_0, v_1, \ldots, v_l)$ of maximal length. If $s \in S$ equals v_0 then $v_i = s^{2^i}$ for $1 \leq i \leq l$. All these powers of s are distinct since they label different vertices in the path. Moreover, $s^{2^{l+1}}$ is a vertex in the cycle of the same connected component as the path, because of the maximality of l.

Consider next the monogenic semigroup generated by s and its index m and period r. It shall be shown that 2^l+1 is a lower bound for the size of the monogenic semigroup, which equals $m + r - 1$ according to (1.3). This is true if $m \geq 2^l + 1$; thus assume that $m \leq 2^l$. Let c denote the length of the cycle in the connected component. Then squaring $s^{2^{l+1}}$ repeatedly yields the element itself after c times; in other words $(s^{2^{l+1}})^{2^c} = s^{2^{l+1+c}} = s^{2^{l+1}}$. Hence r divides $2^{l+1+c} - 2^{l+1}$ or, reformulated, r divides $2^{l+1}(2^c - 1)$. However, if r was to divide $2^l(2^c - 1)$, then $(s^{2^l})^{2^c} = s^{2^{l+c}} = s^{2^l}$ would follow, making s^{2^l} a vertex in the cycle. Therefore $r \geq 2^{l+1}$, which shows that, in any case, $m + r - 1 \geq 2^l + 1$.

If the cycle is a loop, that is if $c = 1$, then repeated squaring for any power of s will eventually equal $s^{2^{l+1}}$ as $(s^j)^{2^{l+1}} = s^{j2^{l+1}} = s^{2^{l+1}}$ for any $j \in \mathbb{N}$. Thus the corresponding vertices have to be in the same connected component as s.

Hence, in the general case, the number of elements in $\langle s \rangle$ is restricted by n, and is restricted by the number of elements in the connected component if the cycle is a loop. Rearranging $2^l + 1 \leq n$, respectively $2^l + 1 \leq k$, yields the bounds from the lemma. □

The graph corresponding to the diagonal of a monogenic semigroup $\langle a \rangle$ with index n and period 1 is connected and has a loop as cycle. The path starting at a and ending at the root of the tree connected to the loop has length $\lceil \log_2(n) \rceil$ which equals $1 + \lfloor \log_2(n-1) \rfloor$ for $n \geq 2$. This shows that the bounds given in Lemma 3.2.4 are tight. The semigroup $\langle a \rangle$ is n-nilpotent according to Lemma 2.1.5. With the previous result many diagonals are identified not to appear in any table defining an associative multiplication. It can be improved further in the restricted situation of tables defining nilpotent semigroups with specified nilpotency rank. To see this, a necessary condition for a diagonal to appear in the multiplication table of a nilpotent semigroup is established. It restricts the *height* of a rooted tree, that is the maximal length of any path, in the graph corresponding to the diagonal.

Lemma 3.2.5 *Let S be a nilpotent semigroup of rank r. Then the graph Γ corresponding to the diagonal of the multiplication table is connected and its cycle is a loop. Moreover, the length of a path in the rooted tree connected to the loop is bounded by $\lceil \log_2(r) \rceil$.*

Proof: Let z denote the zero element in S and let $n = |S|$. If $s \in S$ then $s^k = z$ for all $k \geq r$, in particular for $k = 2^n$. Thus the vertices labelled s and z lie in the same component of Γ. Since s was arbitrary, it follows that Γ is connected and has a loop formed by the edge (z, z).

Let l be the length of a path in the tree rooted at z. Then there exists an $s \in S$ with $s^{2^{l-1}} \neq z$. It follows that $2^{l-1} < r$. Reformulating yields $h \leq \lceil \log_2(r) \rceil$. □

Diagonals for which the corresponding graph has one connected component do not only occur in the case of nilpotent semigroups. The smallest example is given by the cyclic group with two elements, which has a constant function as diagonal.[4] Then again, not every semigroup with one idempotent yields a graph with only one component. A result on how the partitioning into components of the graph influences the structure of the semigroup is given in the following. Remember that the set of idempotent elements in a semigroup S is denoted by $E(S)$, and $K(e)$ denotes the set $\{s \mid s^i = e \text{ for some } i \in \mathbb{N}\}$ for $e \in E(S)$.

Lemma 3.2.6 *Let S be a finite semigroup. Then the partition of S defined by the connected components of the graph Γ of the diagonal of the multiplication table is a refinement of $\{K(e) \mid e \in E(S)\}$. The two partitions are equal if and only if the cycle of each connected component in Γ is a loop.*

Proof: Let e be an idempotent and $s \in K(e)$ be an element with $s^k = e$ for some $k \in \mathbb{N}$. If $t \in S$ labels a vertex in the same connected component of Γ as s, then

[4] Indeed, every finite 2-group – that is, a group in which the order of every element is a power of 2 – leads to a graph with one component, since repeated squaring eventually yields the identity element of the group.

there exist $i, j \in \mathbb{N}$ such that $s^{2^i} = t^{2^j}$. Hence $t^{k2^j} = s^{k2^i} = e^{2^i} = e$, which shows $t \in K(e)$. That idempotents correspond to vertices with a loop completes the proof. $\square$

If the digraph corresponding to a diagonal contains not only loops as cycles and there is more than one idempotent, then it is not possible to deduce from the graph in which way $\{K(e) \mid e \in E(S)\}$ was refined in the previous lemma. Although, the fact that elements in cycles have to belong to *some* set $K(e)$ can be exploited, as the next result demonstrates.

Lemma 3.2.7 *Let S be a finite semigroup and Γ the graph corresponding to the diagonal of the multiplication table of S. Then elements of S labelling vertices in the same cycle of Γ lie in a common subgroup of S. If the length of the cycle is c, then the order of the group elements divides $2^c - 1$, but does not divide $2^k - 1$ for any $k < c$.*

Proof: Let s be a vertex in a cycle of length c. Consider the monogenic semigroup generated by s. It is a group, since $s = s^{2^c}$ shows that the index of $\langle s \rangle$ is 1. From this equation it follows as well that the order of s has to divide $2^c - 1$. On the other hand, it cannot divide $2^k - 1$ for any $k < c$, as otherwise the cycle would have length at most k. $\square$

Due to the result in the previous lemma something can be said about the Green's structure of the semigroup if the graph of its diagonal has a cycle, which is not a loop. Each subgroup of a semigroup is an $\mathcal{H}$-class (see [How95, Theorem 2.2.5]).

Lemma 3.2.7 allows one to identify further diagonals that do not appear in the multiplication table of a semigroup. For cycles of small size Table 3.4 lists possible orders of the group elements. The order of any element must obviously not be greater than the size of the whole semigroup. Moreover, a cyclic group of given order yields a specific graph, which then has to be a subgraph of the graph corresponding to the diagonal. The graph of C_7, for example, consists of one loop and two cycles of length three. Thus if the graph of a diagonal contains one cycle of length 3 it contains in fact at least two.

The restrictions on the height of rooted trees and on possible combinations of cycles strongly suggest that most diagonals cannot appear in the multiplication

Table 3.4 Possible orders of elements labelling vertices in a cycle

length of cycle	orders of elements in cycle
1	1
2	3
3	7
4	5, 15
5	31
6	9, 21, 63
7	127
8	17, 51, 85, 255
9	511

table of a semigroup. This is supported by the empirical evidence in Table 3.3 and the enumeration of rooted trees by height [Rio60].

Question 3.2.8 Are asymptotically all diagonals excluded from appearing in the multiplication table of a semigroup?

Even though there exist results on the asymptotic behaviour of digraphs representing functions [Mut88], a proof to answer Question 3.2.8 in the positive does not seem straightforward.

3.2.2 Allowed diagonals

This subsection is about diagonals that appear in the multiplication table of a semigroup. They will often be referred to as *allowed diagonals*.

The construction of 3-nilpotent semigroups in the previous chapter was such that the products of generators can equal any non-generator. This holds in particular for squares of generators. Hence, any diagonal which fulfils the condition in Lemma 3.2.5 for nilpotency rank 3 does appear in a semigroup. In a similar way the diagonals from monogenic semigroups can be obtained constructively. The n non-equivalent, monogenic semigroups of size n are, for example, generated by the transformations

$$\begin{pmatrix} 1 & 2 & 3 & \cdots & n+1 \\ i & 1 & 2 & \cdots & n \end{pmatrix} \text{ with } 1 \leq i \leq n.$$

Here i is the period of the semigroup.

In every semigroup each element generates a monogenic subsemigroup. The diagonal of a finite semigroup S can thus be considered as the union of the diagonals of the monogenic subsemigroups of S. Consider for example the semigroup defined by the multiplication table in Figure 3.2. For three of its monogenic subsemigroups containing together all elements in the semigroup the digraphs corresponding to the respective diagonals of their multiplication tables are drawn in Figure 3.4. They merge to the graph corresponding to the diagonal of the whole semigroup.

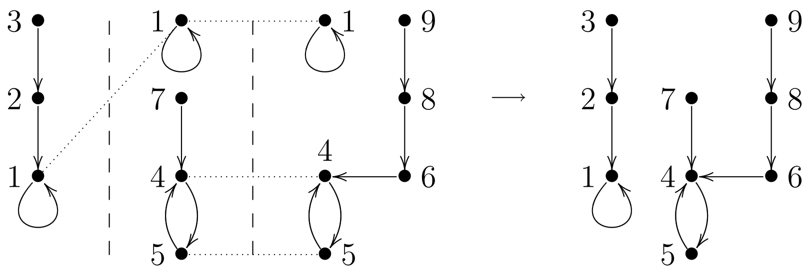

Figure 3.4 Assembling the diagonal $(1, 1, 2, 5, 4, 4, 4, 6, 8)$ of the semigroup from Figure 3.2 using its monogenic subsemigroups

Assembling graphs of monogenic semigroups in arbitrary ways does not always lead to the diagonal of a semigroup. Merging, for example, the first and the third graph in Figure 3.4 yields the graph corresponding to a diagonal that does not appear in any multiplication table of a semigroup with 8 elements.

Methods to construct semigroups from smaller semigroups shall be used to deduce how to assemble graphs corresponding to diagonals of semigroups into larger diagonals appearing in a semigroup.

Lemma 3.2.9 *Let S and T be semigroups on disjoint sets with graphs Γ and Δ corresponding to their diagonals. Then $\Gamma \cup \Delta$ is the graph of the diagonal of a semigroup.*

Proof: Define a multiplication on the union of the elements in S and T as follows. Products of elements, both in one of S or T are evaluated in the respective semigroup. For $s \in S$ and $t \in T$ define $st = ts = s$. This yields an associative multiplication as in every product with at least one factor in S, factors from T are ignored. The diagonal of the multiplication table equals $\Gamma \cup \Delta$. $\square$

The previous result deals with the simplest case of disjoint graphs. The situation becomes more complicated if graphs of allowed diagonals merge to give another allowed diagonal, as shown in Figure 3.4. In the following result one of the graphs involved has to arise from a semigroup containing a zero element.

Lemma 3.2.10 *Let S and T be semigroups such that $S \cap T = \{e\}$, where e is an idempotent in S and a zero element in T. If Γ and Δ denote the graphs corresponding to the diagonals of the multiplication tables of S and T respectively, then there exists a semigroup with $\Gamma \cup \Delta$ corresponding to the diagonal of its multiplication table.*

Proof: Define a multiplication on the union of the elements in S and T as follows. Products of elements, both in either S or T are evaluated in the respective semigroup. For $s \in S$ and $t \in T$ define $st = se$ and $ts = es$, evaluated in S. Note that the multiplication is well-defined as the only element S and T have in common is an idempotent in both semigroups and a zero element in T. The multiplication is associative, essentially because in every mixed product factors from T are replaced by the same element from S. Moreover, if $s \in S$ and $t_1, t_2 \in T$ then $s(t_1t_2) = (se)e = (st_1)t_2$, in which case it is needed that e is an idempotent element. The semigroup constructed in this way has $\Gamma \cup \Delta$ corresponding to the diagonal of its multiplication table. □

In the next two results, allowed graphs are enlarged by adjoining one or two new elements to a semigroup. While this can still be seen as taking the union of graphs belonging to smaller semigroups, it will be presented in a different way.

Lemma 3.2.11 *Let S be a finite semigroup and Γ the graph corresponding to the diagonal of its multiplication table. For a vertex $v \in \Gamma$ construct $\Gamma_{v,\mathrm{I}}$ by adding a new vertex to Γ and an edge from the new vertex to v. If v is the end vertex of an edge in Γ then $\Gamma_{v,\mathrm{I}}$ is a graph corresponding to the diagonal of a semigroup.*

Proof: Let $t \in S$ such that $t^2 = v$ and denote by x an element not in S. Extend the multiplication of S to $S \cup \{x\}$ by defining $xs = ts$ and $sx = st$ for all $s \in S$, and $x^2 = t^2$. As every appearance of x in a product is substituted by t, the multiplication defined on $S \cup \{x\}$ is associative. The graph corresponding to the

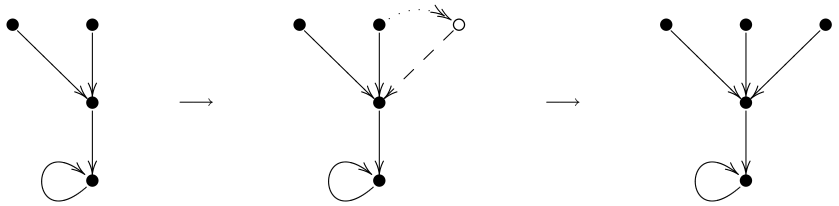

Figure 3.5 Replication of one edge in a digraph

diagonal of the new semigroup is $\Gamma_{v,\mathrm{I}}$. □

The construction in the previous lemma is illustrated in Figure 3.5.

Lemma 3.2.12 *Let S be a finite semigroup and Γ the graph corresponding to the diagonal of its multiplication table. For a vertex $v \in \Gamma$ construct $\Gamma_{v,\mathrm{II}}$ by adding two new vertices to Γ, an edge from one to the other and an edge from the end vertex of the new edge to v. If v is the end vertex of an edge for which the start vertex is itself the end vertex of an edge in Γ then $\Gamma_{v,\mathrm{II}}$ is a graph corresponding to the diagonal of a semigroup.*

Proof: Let $s \in S$ such that $s^4 = v$ and denote with x, y two elements not in S. Extend the multiplication of S to $S \cup \{x, y\}$ by defining $x^2 = y$ and substituting in any other product s for x and s^2 for y. When products with three elements are evaluated all appearances of x and y will eventually be substituted and the substitution complies with the equality $x^2 = y$. Hence the multiplication is associative. As $x^2 = y$ and $y^2 = v$ the graph corresponding to the diagonal is $\Gamma_{v,\mathrm{II}}$. □

The last result means that replicating a walk of length two in the graph of an allowed diagonal yields a new allowed diagonal. The process might involve parts of a cycle, as the example in Figure 3.6 illustrates.

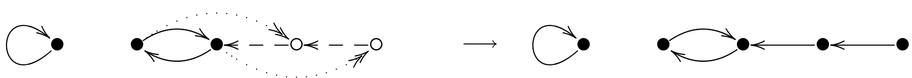

Figure 3.6 Replication of two edges in a digraph

Starting with the diagonals from monogenic semigroups and using the results in this section one can build a set of allowed diagonals for any order. To use Lemma 3.2.10 note that a monogenic semigroup with zero element is nilpotent due to Lemma 2.1.1. According to Lemma 2.1.5 there is one such semigroup of every order, characterised by the fact that its period is 1. A nilpotent, monogenic semigroup of order n is, for example, generated by the transformation

$$\begin{pmatrix} 1 & 2 & 3 & \cdots & n+1 \\ 1 & 1 & 2 & \cdots & n \end{pmatrix}.$$

Building allowed diagonals in this way yields, in fact, all diagonals that appear in a semigroup for orders up to 8. For order 9 all but two allowed diagonals are constructed by this process. The graphs in Figure 3.7 illustrate the two diagonals missed. One of them consists of one loop and four cycles of length 2. According to Lemma 3.2.7 a semigroup with this diagonal must be a group. Extending the process by adding all diagonals which appear in groups would take care of this and similar diagonals – assuming that the groups of the given order, or at least their diagonals, are known. Still, the diagonal featured in Figure 3.4 would not show up. It seems like there is no general rule that would allow one to construct the semigroup with this diagonal (see Figure 3.2) from a smaller semigroup. It is conceivable that there are more exceptions for higher orders. At least for order 10 though (the highest order which was examined) there is no new exception. That is, when starting with all allowed diagonals of orders 1 to 9 and the diagonals of monogenic semigroups of order 10, the construction rules from this section yield all allowed diagonals of order 10.

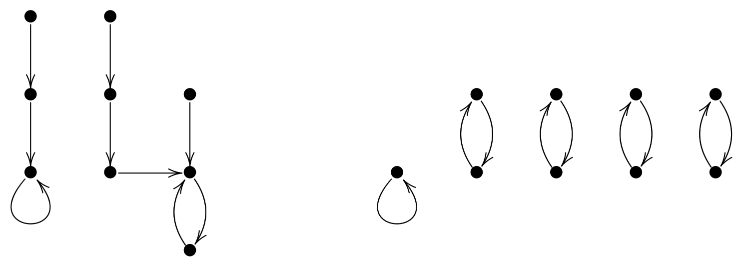

Figure 3.7 The two graphs corresponding to diagonals of semigroups of order 9 not found with the described construction

All new semigroups constructed from smaller ones using the results in this subsection are commutative, if one starts with monogenic semigroups. Indeed, up to order 10 the experimental results in Table 3.3 showed that all allowed diagonals appear in commutative semigroups.

Question 3.2.13 Does each allowed diagonal appear in a commutative semigroup?

If the answer to this question was 'no', it would be a fundamental problem for the presented process of finding allowed diagonals. A 'yes' as answer to the following question would have a similarly negative effect.

Question 3.2.14 Can a connected component, which has a loop as cycle and does not correspond to an allowed diagonal itself, appear in the graph of an allowed diagonal?

While on the face Questions 3.2.13 and 3.2.14 look rather different, they are in fact closely connected. In an allowed diagonal that appears in a commutative semigroup, elements, labelling the vertices in a connected component having a loop, form a subsemigroup. This is shown by the following lemma.

Lemma 3.2.15 *Let S be a commutative semigroup and let $e \in E(S)$. Further let $x, y \in S$ be elements in the connected component containing e of the graph corresponding to the diagonal of S. Then $z = xy = yx$ is in the same connected component.*

Proof: Take $m \in \mathbb{N}$ such that $x^{2^m} = y^{2^m} = e$. Then $z^{2^m} = x^{2^m} y^{2^m} = e^2 = e$. $\square$

Note that answering Question 3.2.14 in the negative would as well give a very useful criterion to exclude diagonals as discussed in the previous subsection.

The reader might expect that criteria from the last two subsections, deciding whether a diagonal appears in the multiplication table of a semigroup, play a role in the computer search for semigroups. It will turn out that this is not the case, at least not for the methods presented in the forthcoming chapters. Observations for the diagonals that appear in the multiplication table of a monoid had been utilised in [DK08], though the approach has already been superseded [DK09].

4 Semigroups of Order at most 8

Prior to this work the number of non-equivalent semigroups was known up to order 8. Forsythe introduced computer search to the enumeration of semigroups when he programmed SWAC to count the 126 distinct semigroups with 4 elements [For55].[1] To obtain the results for subsequent orders, various authors implemented specialised programs [MS55, Ple67, JW77, SYT94].

In this chapter the known results are reproduced using a new approach in the enumeration of algebraic objects utilising constraint satisfaction (an area in computer science concerned with combinatorial problem solving). This allows one to use highly developed existing software instead of implementing a specialised program. The new approach, including a realisation of it, is introduced and explained in detail in the first section of this chapter.

The main reason for the reproduction of the enumeration results was to obtain the semigroups of order at most 8 and to use them for the creation of an electronic data library. The library contains the semigroups as well as information about them and is available as GAP [GAP08] package Smallsemi [DM10]. The construction of Smallsemi is subject of Section 4.2.

4.1 Enumeration Using Constraint Satisfaction

Constraint satisfaction is an area in artificial intelligence concerned with modelling and solving a wide range of combinatorial problems. Standard examples of problems are scheduling and planning. While formulations and theoretical research follow a strict mathematical definition, the practical side of constraints satisfaction combines search strategies with propagation, largely using heuristic methods. The aim is to provide convenient tools – so called solvers – suitable for many dif-

[1] Tamura had already determined the semigroups of order 4 by hand calculations [Tam54].

ferent applications. To get information about research in constraint satisfaction, the reader might want to start with [RvBW06].

This section contains information on a basic approach to the computational enumeration of semigroups up to equivalence utilising constraint satisfaction. Known enumeration results from [SYT94] are reproduced. Basic definitions from constraint satisfaction are provided.

Definition 4.1.1 A *constraint satisfaction problem (CSP)* is a triple (V, D, C), consisting of a finite set V of *variables*, a finite set D, called the *domain*, of *values*, and a set C containing subsets of D^V (that is, all functions from V to D) called *constraints*.

Definition 4.1.1 gives a rigorous description of a CSP that is not very useful in practice. Most important, instead of being subsets of D^V, constraints are formulated as conditions uniquely defining such subsets. While this is not an important distinction for the theory of constraint programming, it is when a CSP is used to actually solve a specific problem.

Intuitively it is clear that one is looking for assignments of values in the domain of a CSP to all variables such that no constraint is violated. This is formalised in the next definition.

Definition 4.1.2 Let $L = (V, D, C)$ be a CSP. A partial function $f : V \to D$ is an *instantiation*. An instantiation f *satisfies* a constraint, if there exists a function F in the constraint, such that $F(x) = f(x)$ for all $x \in V$ on which f is defined. An instantiation is *valid*, if it satisfies all the constraints in C. An instantiation defined on all variables is a *total* instantiation. A valid, total instantiation is a *solution* to L. The number of all solutions of L will be denoted by $|L|$.

The rest of this section is divided into five parts explaining step by step how semigroups are enumerated using a CSP. First, the problem is formulated as a CSP. This is done for the enumeration of all different semigroups on $[n]$ in the forthcoming subsection. To count the number of structurally different semigroups on n elements, the initial CSP is extended in Subsection 4.1.2 using a well-known technique to eliminate symmetries in CSPs. This concludes the theoretical considerations. In Subsection 4.1.3 the CSP is translated into input for a constraint solver – software that is designed to return solutions to a CSP. This step is the closest to programming in the described approach. The choices made for the input can drastically influence the time a solver takes to return the solutions. Changes to constraints that do not change the solutions of the CSP are discussed in Subsection 4.1.4. Finally, the results from solving the input to reproduce the known enumeration of semigroups are presented in the last subsection.

4.1.1 Formulation of the basic CSP

As a first example for the usage of constraint satisfaction in the enumeration of semigroups, all the different semigroups on the set $[n]$ shall be determined. Solving a problem using constraint satisfaction involves several steps. It starts with the formulation of the problem as a CSP.

CSP 4.1.3 For $n \in \mathbb{N}$ define a CSP $L_n = (V_n, D_n, C_n)$. The set V_n consists of n^2 variables $\{T_{i,j} \mid 1 \leq i, j \leq n\}$, one for each position in an $(n \times n)$-multiplication table, having domain $D_n = [n]$. The constraints in C_n are

$$T_{T_{i,j},k} = T_{i,T_{j,k}} \text{ for all } i, j, k \in [n], \tag{4.1}$$

reflecting associativity. (Note that (4.1) is a slight misuse of notation. Using a variable as index shall refer to its value.)

It is straightforward to verify that the multiplication table defined by a solution of L_n from CSP 4.1.3 will be associative, and that, in turn, the table of every associative multiplication fulfils the constraints in C_n. Thus the valid full assignments for L_n correspond to the semigroups on $[n]$. As the constraints C_n enforcing associativity will be present in every following model, the solutions will always define semigroups and are often referred to as such.

The number of all different semigroups on $[n]$ grows rapidly with n and most of the semigroups are 3-nilpotent [KRS76]. As it is known how to construct the 3-nilpotent semigroups on $[n]$ (see Section 2.3), they do not have to be searched for. Adding the constraint

$$\exists i, j, k, r, s, t \in [n] : T_{i,T_{j,k}} \neq T_{r,T_{s,t}} \tag{4.2}$$

to C_n yields the CSP L_n^{-3}, having all different semigroups on $[n]$ as solutions which are neither 3-nilpotent nor a zero semigroup.

4.1.2 Breaking symmetries

Recall that the actual aim is not to find all semigroups on $[n]$, but to find one semigroup of every equivalence class under the action of $S_n \times C_2$, that is, under isomorphism and anti-isomorphism. It is very common that modelling a problem as a CSP introduces symmetries. Here this happens because of the representation of semigroups by their multiplication table. For this purpose identifiers, 1 up to n, were introduced for the n elements that are initially indistinguishable. Moreover, the model fixes the direction to read the multiplication off the table.

Starting from the solutions of L_n one can identify equivalent ones in a post-process and thus obtain semigroups of order n up to equivalence. Due to the large number of semigroups on $[n]$ this is impractical. If one is interested in the semigroups up to equivalence, it will be far more efficient to extend L_n with constraints which ensure that only one semigroup per equivalence class is a solution. The so called 'lex-leader' approach is a well-known technique for this purpose [CGLR96]. Some preparation is needed before it is explained.

Definition 4.1.4 Let $L = (V, D, C)$ be a CSP.

(i) Elements in the set $V \times D$ are called *literals*. Literals are denoted in the form $(x = k)$ with $x \in V$ and $k \in D$.

(ii) Let χ denote the set of all literals of L. A permutation $\pi \in S_\chi$ is a *symmetry* of L if, under the induced action on subsets of χ, instantiations are mapped to instantiations and solutions to solutions.

(iii) A *variable-value symmetry* is a symmetry $\pi \in S_\chi$ such that there exists an element (τ, δ) in $S_V \times S_D$ with $(x = k)^\pi = (x^\tau = k^\delta)$ for all $(x = k) \in \chi$.

The given definition of symmetry of a CSP is relatively strong. On the other hand, only symmetries of L_n induced by the action of $S_n \times C_2$ on multiplication tables are of interest here. All these symmetries are variable-value symmetries and any variable-value symmetry will always send instantiations to instantiations. Note that indeed every element in $S_n \times C_2$ induces a symmetry as every solution of L_n – that is, any associative multiplication table – is mapped to a solution. For more information on symmetries in CSPs, including different definitions see [RvBW06, Chapter 10].

The symmetries of a CSP $L = (V, D, C)$ form a group G. Two solutions of L are *symmetric*, if they are G-equivalent under the induced action of G on subsets of the literals. The idea of lex-leader is to order solutions by defining an order on the literals of the CSP. This allows one to define the canonical representative in each orbit of symmetric solutions of L to be the solution which is smallest (or largest) with respect to the order. To define an order on solutions of L, first fix an ordering $(\chi_1, \chi_2, \ldots, \chi_{|V||D|})$ of the literals $\chi = V \times D$. Given the fixed ordering of the literals, an instantiation can be represented as a bit vector of length $|V||D|$. The bit in the i-th position is 1 if χ_i is contained in the instantiation and otherwise the bit is 0. The bit vector for the instantiation $I \subseteq \chi$ corresponding to the ordering of the literals $(\chi_1, \chi_2, \ldots, \chi_{|V||D|})$ will be denoted by $(\chi_1, \chi_2, \ldots, \chi_{|V||D|})_{|I}$. Of all bit vectors corresponding to the elements in an orbit of symmetric solutions in L, one is the lexicographic maximal, which shall be the property identifying the canonical solution. If $\geq_{\text{lex}}$ denotes the standard lexicographic order on vectors, extend L by adding, for all $\pi \in G$, the constraint

$$(\chi_1, \chi_2, \ldots, \chi_{|V||D|})_{|I} \geq_{\text{lex}} (\chi_1^\pi, \chi_2^\pi, \ldots, \chi_{|V||D|}^\pi)_{|I}. \tag{4.3}$$

Then, from each set of symmetric solutions in L, exactly those with lexicographic greatest bit vector are solutions of the extended CSP.

The lex-leader method works as well for subgroups of the symmetries of a CSP. In the case of L_n (respectively L_n^{-3}) the solutions form orbits under $S_n \times C_2$. Adding Constraint (4.3) for each element in $S_n \times C_2$ gives a new CSP, which has as solutions one semigroup from every equivalence class and is denoted by $\overline{L}_n$ (respectively $\overline{L}_n^{-3}$).

4.1.3 Instances from L_n and $\overline{L}_n$

After modelling a problem as a CSP it has to be translated into input for a constraint solver. The availability of types of values for variables and, in particular, of constraints varies from program to program. The solver used in the work presented here is Minion [GJM06]. An input file for a solver will be referred to as *instance*.

Regarding the needs to input the CSP L_n into Minion, matrices exist as variable types (in the form of an array of arrays) and an interval of integers is a possible domain.[2] On the other hand, associativity is not a common constraint in constraint satisfaction and hence not directly supported. This shortcoming can be circumvented by introducing one auxiliary variable (a variable which is not part of the formulation of the problem) for each equation $T_{T_{i,j},k} = T_{i,T_{j,k}}$ from (4.1). Using two separate constraints, the auxiliary variable is then required to equal both the left and the right hand side of the equation.

To create the lex-leader constraints in $\overline{L}_n$ more auxiliary variables are needed. One Boolean variable for each literal in L_n is introduced. The Boolean variable corresponding to the literal $(T_{i,j} = k)$ for $i, j, k \in [n]$ is 'true' if $T_{i,j}$ has value k and it is 'false' if $T_{i,j}$ has another value.

When creating the input, certain choices are made that can influence the efficiency of the search drastically. The choices made for L_n and $\overline{L}_n$ are described in the following.

Variable order Using the adaptation by Satoh et al. of Plemmons' idea, the chosen search order for the variables in L_n puts the diagonal positions first and proceeds row by row with the remaining positions. That is, the search order is given by the tuple

$$(T_{1,1}, T_{2,2}, \ldots, T_{n,n}, T_{1,2}, \ldots, T_{1,n}, T_{2,1}, T_{2,3}, \ldots, T_{2,n}, \ldots, T_{n,1}, \ldots, T_{n,n-1}). \tag{4.4}$$

Value order The value order is chosen to be ascending for each variable in L_n.

Literal order This is only relevant for $\overline{L}_n$. For best performance of the lex-leader constraints the ordering of the literals must be in line with the search order for variables and values.

Different implementations For some constraints different implementations are provided. Which implementation is more efficient relies on the rest of the problem. This is relevant in the case of the lexicographic comparison of vectors used for the lex-leader constraints, which has three different implementations in Minion.

[2]Note that it is standard in computer science to start counting with 0. Hence, the set $[n]$ will be represented in Minion by $\{0, 1, \ldots, n-1\}$. As this is a minor technical detail it will be mostly ignored for further considerations.

Different constraints Some conditions can be expressed using various constraints. This is important for the constraint in (4.2) to forbid 3-nilpotent solutions. The constraint says, there have to exist two of the auxiliary variables, introduced to require associativity, with different values. This is guaranteed by adding n constraints, one for each possible value $1 \leq i \leq n$, forbidding the vector of all auxiliary variables to equal the constant vector of length n^3 with value i. The same effect could be achieved using an occurrence constraint, which restricts the occurrence of a certain value in a vector to a fixed number. Restricting the number of occurrences of each value i to $n^3 - 1$ would prevent the vector from being constant.

Having the order of the literals determined by the search order means that the n^3 literals appear in n^2 blocks of the form $((T_{i,j} = 1), (T_{i,j} = 2), \dots, (T_{i,j} = n))$ for $1 \leq i, j \leq n$ in the ordering. With the chosen ordering the solution tables will be minimal in their orbit. It is easy to see that the maximal bit vector corresponding to the literals arises from the table minimal in its equivalence class. If T and U are two tables with $T < U$ and (i, j) is the first position (with respect to the search order) in which they differ, then $T_{i,j} < U_{i,j}$. Hence, in the bit vectors for T and U the 1 in the block corresponding to position (i, j) appears earlier for T than for U.

4.1.4 Optimising constraints

Any vector that appears in one of the lex-leader constraints (4.3) maintains the block structure in the ordering of the literals discussed at the end of the previous subsection. The structure is preserved since all symmetries are variable-value symmetries and hence the images of literals containing the same variable will still contain the same variable. In any total instantiation one of the Boolean variables corresponding to $\{(T_{i,j} = 1), (T_{i,j} = 2), \dots, (T_{i,j} = n)\}$ will be 'true' and the other $n - 1$ will be 'false'. Thus the value of the least significant literal in every block never decides about the lexicographic order of the vectors in (4.3) and can therefore be removed. There is a simple, general rule to reduce the length of a lex-leader constraint.

Remark 4.1.5 Consider the constraint

$$(\chi_1, \chi_2, \dots, \chi_m)_{|I} \geq_{lex} (\chi'_1, \chi'_2, \dots, \chi'_m)_{|I}. \tag{4.5}$$

If for every instantiation I with $(\chi_1, \ldots, \chi_{k-1})_{|I} = (\chi'_1, \ldots, \chi'_{k-1})_{|I}$ the equality $(\chi_k)_{|I} = (\chi'_k)_{|I}$ holds, then (4.5) may be replaced by

$$(\chi_1, \ldots, \chi_{k-1}, \chi_{k+1}, \ldots, \chi_m)_{|I} \geq_{lex} (\chi'_1, \ldots, \chi'_{k-1}, \chi'_{k+1}, \ldots, \chi'_m)_{|I}.$$

The easiest example for an application of Remark 4.1.5 occurs for a symmetry π which fixes some literal. If $\chi_i = \chi_i^\pi$, then no decision on the lexicographic order is made in bit i. The next step is to look at a cycle of length two in π. If $\chi_i = \chi_j^\pi$ and $\chi_j = \chi_i^\pi$ with $i < j$ then the j-th position is not significant for the lexicographic comparison. Either the lexicographic ordering is decided before the j-th position or otherwise $\chi_i = \chi_i^\pi$ and hence $\chi_j = \chi_i^\pi = \chi_i = \chi_j^\pi$. This example given for a transposition in π generalises to cycles of arbitrary length. The literal of a cycle corresponding to the least significant bit will never decide which vector is lexicographically greater.

In general, methods that reduce the number and length of lex-leader constraints without changing the set of solutions tend to be costly or are not likely to give an essential reduction. In fact, there are problem classes where an exponential number of lex-leader constraints is required [LR04].

The idea of changing constraints without changing the solutions shows up in two ways. A constraint in a CSP is *redundant* if removing it does not change the set of solutions. A constraint is *implied* in a CSP if it is not part of the CSP, but adding it does not change the set of solutions. Obviously, the distinction between redundant and implied constraints is somewhat artificial, and is indeed not used consistently in the literature. In practice, a redundant constraint will be one that the CSP does not benefit from for the purpose of solving it. Hence removing the constraint is an improvement to the model. On the other hand, adding an implied constraint aims to make solving the CSP easier.

Redundant and implied constraints will become more important for CSPs introduced later. Here only one implied constraint for L_n is mentioned. Every finite semigroup has an idempotent (see Remark 1.2.3), which yields the implied constraint

$$\exists i : T_{i,i} = i. \tag{4.6}$$

This is a very simple example demonstrating how an implied constraint rises from mathematical knowledge rather than being deduced directly from the constraints already present.

Of particular interest is the interaction of the constraint in (4.6) with the lex-leader constraints from (4.3) in $\overline{L}_n$. The first literal in the ordering is $(T_{1,1} = 1)$. Under the permutation $(1\,i)$ the first literal becomes $(T_{i,i} = i)$. Thus, since for some $1 \leq i \leq n$ the equality $T_{i,i} = i$ holds, $T_{1,1} = 1$ must hold in every canonical solution. Hence (4.6) can be replaced by $T_{1,1} = 1$. Even though the latter rules out many instantiations as solutions – implicitly identifying them as not canonical – it does not avoid posting all $2n!$ lex-leader constraints. If $T_{i,i} = i$ holds for all $1 \leq i \leq n$, as it does for bands, all images of a solution table satisfy $T_{1,1} = 1$. An important point to note is that many elements in $S_n \times C_2$ are no longer symmetries of the CSP, if one adds $T_{1,1} = 1$ as constraint. The group of symmetries was used to introduce the lex-leader method, but the aim of adding the constraints (4.3) to the CSP L_n is to get one solution from every equivalence class under the action of $S_n \times C_2$. That the latter is not a subgroup of the symmetries any more does not matter: what is important is that each canonical solution fulfils the new constraint and hence stays a solution.

The fact that $T_{1,1} = 1$ holds, influences as well constraint (4.2) in $\overline{L}_n^{-3}$ to eliminate nilpotent semigroups of rank at most 3 from the solution set. As 1 is an idempotent, it is the only candidate for the zero in a nilpotent semigroup. It therefore suffices to forbid the vector of auxiliary variables, representing all products of length three, to equal the constant vector with value 1; or to require that 1 occurs at most $n^3 - 1$ times in that vector. This allows as well a third formulation of (4.2). Requiring the sum of the vector containing the auxiliary variables to equal at least $n^3 + 1$, prevents it from being a constant vector with 1 as entry. Since this formulation involves summation of the entries of the vector, it tends to be less efficient that the realisations described in Subsection 4.1.3 and is not used.

4.1.5 Computations for L_n and $\overline{L}_n$

The input for Minion was produced using an interface from GAP written by Linton, which is available on the attached DVD in the file `minion.g` (see Appendix C). The interface was used to create Minion instances for L_n and L_n^{-3} with $1 \leq n \leq 7$ as well as for $\overline{L}_n$ and $\overline{L}_n^{-3}$ with $1 \leq n \leq 8$. The instances can also be found on the DVD. If one is not familiar with the input language for Minion, it will be difficult

Table 4.1 Enumeration of all different semigroups on $[n]$

n	1	2	3	4	5	6	7
L_n, solutions	1	8	113	3 492	183 732	17 061 118	7 743 056 064
–, solve time	ϵ	ϵ	ϵ	ϵ	4 s	334 s	73 563 s
L_n^{-3}, solutions	0	6	104	3 308	172 007	13 971 862	1 798 975 985
–, solve time	ϵ	ϵ	ϵ	ϵ	3 s	392 s	115 311 s

The times are rounded to seconds. They were obtained using the 64-bit executable of Minion version 0.9 on a machine with 2.66 GHz Intel X-5430 processor and 16 GB RAM. The symbol ϵ stands for a time less than 0.5 s.

to understand the instances. As an example of a CSP turned into input for Minion, an annotated version of the instance for $\overline{L}_2^{-3}$ is given in Figure 4.1.[3] Be aware that this instance and every instance on the DVD use an obsolete syntax. The current version 0.9 of Minion accepts these instances as input, but describes a different syntax in its documentation.

Solving the created instances yields the results in Table 4.1 and Table 4.2. Timings were obtained from a single run, while the results were checked in a second run. The numbers for $|L_n|$ and $|\overline{L}_n|$ match the known numbers of semigroups. The differences $|L_n| - |L_n^{-3}|$ and $|\overline{L}_n| - |\overline{L}_n^{-3}|$ coincide with the numbers of semigroups of nilpotency rank at most 3 (see Appendix A.1).

It might be surprising at first that the smaller problems L_n^{-3}, not including semigroups of nilpotency rank at most 3, are solved slower than L_n for the same n according to the timings in Table 4.1. This makes more sense if one remembers how constraint (4.2) is expressed in this case: the vector of all products of three elements is considered. To verify that this vector is not constant, the whole table has to be known. This shows an essential problem if knowledge about a certain class of semigroups shall be used to simplify the CSP. A simplification, measured in runtime, will only happen if semigroups are excluded using a constraint that propagates well. Even though the constraint to exclude 3-nilpotent semigroups from the solutions is essentially the same for the instances $\overline{L}_n^{-3}$, it seems to propagate better in combination with the lex-leader constraints.

Note that for $\overline{L}_n$ and $\overline{L}_n^{-3}$ for $1 \leq n \leq 7$ the setup of the instance files using GAP takes longer than solving the instance. Nevertheless, the computationally

[3] As mentioned before, $\{0, 1\}$ is used instead of $\{1, 2\}$ in Minion.

```
MINION 1
#autogenerated by GAP
# Find semigroups of order 2
### boolean variables
8 # 8 auxiliary variables, one for each literal
0
0
### integer variables; domain: range from 0 to 1
12 0 1 12 # 4 for table positions, 8 for associativity
0
### search orders
[x8,x9,x10,x11] # variable order
[a,a,a,a] # value order (a = ascending)
5             # 5 vectors
[x0,x2,x4,x6] # reduced vector of literals for lex-leader constraints
[x0,x4,x2,x6] # reduced vector of literals of transposed table
[x7,x3,x5,x1] # ... of table under transposition (0 1)
[x7,x5,x3,x1] # ... of transposed table under transposition (0 1)
### vector of auxiliary variables for associativity
[x12,x13,x14,x15,x16,x17,x18,x19]
1        # 1 matrix
[[x8,x9] # the multiplication table as matrix
,[x10,x11]]
0
objective none
print m0
### connect boolean variables to entries in MT
reify(eq(x8,0),x0) # entry in first position is 0
reify(eq(x8,1),x1) # entry in first position is 1
[...] # analogue for the remaining 3 positions in the table
### constraints enforcing associativity
watchelement([x8,x9],x8,x12)  # auxiliary variable x12 equals 0.(0.0)
watchelement([x8,x10],x8,x12) # ... and as well (0.0).0
watchelement([x8,x9],x9,x13)  # auxiliary variable x13 equals 0.(0.1)
watchelement([x9,x11],x8,x13) # ... and as well (0.0).1
[...] # analogue for the remaining 6 auxiliary variables
### lex-leader constraints
lexleq[quick](v1,v0) # transposed table is less or equal to solution
lexleq[quick](v2,v0) # image under (0 1) is less or equal to solution
lexleq[quick](v3,v0)
### there are at most 7 products of 3 elements equal to 0
occurrenceleq(v4,0,7)
```

Figure 4.1 Minion instance for $\overline{L}_2^{-3}$

Table 4.2 Enumeration of non-equivalent semigroups on $[n]$

n	1	2	3	4	5	6	7	8
$\overline{L}_n$, solutions	1	4	18	126	1 160	15 973	836 021	1 843 120 128
–, setup time	ϵ	ϵ	ϵ	ϵ	ϵ	3 s	90 s	5 457 s
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	2 s	89 s	552 071 s
$\overline{L}_n^{-3}$, solutions	0	3	16	117	1 075	13 312	226 223	11 433 105
–, setup time	ϵ	ϵ	ϵ	ϵ	ϵ	3 s	86 s	5 555 s
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	2 s	61 s	11 119 s

The times are rounded to seconds. They were obtained using the 64-bit executable of Minion version 0.9 on a machine with 2.66 GHz Intel X-5430 processor and 16 GB RAM. The setup was done with GAP. The symbol ϵ stands for a time less than 0.5 s.

difficult part, limiting this approach to $n = 8$, is finding the solutions of the CSP with Minion. Improving the GAP times seemed therefore irrelevant. One simple way of doing so, is not to apply Remark 4.1.5 to the lex-leader constraints (4.3), which, on the downside, slightly increases the time and memory requirements for the Minion computation. The setup time without lex-leader constraints, that is for the CSPs L_n and L_n^{-3}, is negligible.

4.2 A Data Library of Small Semigroups

Plemmons mentioned in [Ple67] that he stored all semigroups of order 6 on magnetic tape. It seems unlikely that many people other than himself had access to the data. Jürgensen and Wick [JW77] state that they did not store the semigroups of order 7 because of their large number, and the same is likely to be true for the semigroups of order 8 found by Satoh et al. [SYT94].

One motivation in reproducing the known enumeration results for semigroups was to actually obtain the semigroups of orders 1 to 8 up to equivalence and to use them for the creation of an electronic database: Smallsemi [DM10] by Mitchell and the author is an extension – a so called package – for the computer algebra system GAP [GAP08]. The integration of the data library into GAP allows one to analyse the semigroups in a convenient way and enables other mathematicians to access them. An ancestor of Smallsemi is available for an earlier version of GAP [S+97]: GLISSANDO [Nöb97] contains semigroups of orders 1 to 5 up to isomorphism and near-rings of orders 2 to 15.[4] The creation of Smallsemi was inspired by other data libraries in GAP, in particular the SmallGroups library [BEO02].

To obtain the semigroups and not just their number using the approach introduced in the previous sections of this chapter is uncomplicated. Depending on the command line switch used to execute Minion the output is the number of solutions or the solutions themselves. Problematic is the amount of data obtained in this way. How the data was compressed to a reasonable size inside the library is explained in the first part of this section. In addition to the semigroups, information about them was computed and included in the library. A list of the precomputed properties is given in Subsection 4.2.2. The last subsection contains some hints on possible usages of Smallsemi. A copy of Smallsemi is included on the attached DVD (Appendix C).

4.2.1 The semigroups in the library

Orders 2 to 7

The semigroups of orders 2 to 7 are produced using the approach from Section 4.1. For each n with $2 \leq n \leq 7$ there is one file containing the data from all tables of semigroups of order n. The data is arranged in the following way. Reading the entries in a table row by row, each table T corresponds to the vector

$$(T_{1,1}, T_{1,2}, \dots, T_{1,n}, T_{2,1}, T_{2,2}, \dots, T_{2,n}, \dots, T_{n,1}, T_{n,2}, \dots, T_{n,n})$$

of length n^2. The entry $T_{1,1}$ always equals 1 in any solution of $\overline{L}_n$ – because every finite semigroup has an idempotent and solutions are minimal in their equivalence class – and is therefore not stored. Subtracting 1 from each entry yields a vector of length $n^2 - 1$ with integer entries in the range 0 to $n - 1$. Since $n \leq 10$, the value of each such integer is a single digit. This makes separators between digits superfluous. Finally the vectors corresponding to the tables are put into the columns of a text file. Hence, one obtains a file with $n^2 - 1$ lines, each line containing one character from $\{0, 1, \dots, n-1\}$ for each semigroups of order n.

[4]While the semigroup library of GLISSANDO was discontinued, the near-rings were ported to the GAP package SONATA [ABE$^+$03].

Example 4.2.1 Consider the four multiplication tables of semigroups of order 2:

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}. \tag{4.7}$$

They lead to the vectors $(1,1,1,1), (1,2,2,1), (1,1,1,2)$, and $(1,1,2,2)$. After removing the first position from each and subtracting 1 from every entry one obtains $(0,0,0), (1,1,0), (0,0,1)$, and $(0,1,1)$. Writing these vectors as columns in a text file yields

```
0100
0101
0011
```

which is exactly the content of the data file in Smallsemi storing the semigroups of order 2.

Note that the multiplication tables are sorted with respect to the ordering of the table positions used in search. Remember that the search order put the diagonal positions first and then the remaining positions row by row, as shown in Equation (4.4). This is the order in which the tables are returned as output by Minion.

The presented way of formatting the data allows fast recovery of the multiplication tables, but is not an efficient way of storing the information. Each file contains only n different characters and a lot of repetition. On the other hand, arranging the data in the way at hand yields a very effective compression of the text files with standard tools. The size of the file for $n = 7$ is $48 \cdot 836022 - 1 = 40\,129\,055$ Bytes; essentially the number of lines times the number of semigroups. Compressing the file with gzip [Deu96] reduces the disc space needed to roughly 619 KB. The original content of files compressed with gzip can be read into GAP without decompressing them – if gzip is present.[5] This approach keeps the recovery of multiplication tables essentially as fast as for uncompressed data, while requiring only a fraction of the disc space. At the same time no specialised method for compression needed to be developed. Nevertheless, the factor of compression is of the same order of magnitude as for the groups of orders 512 and 1536 in the SmallGroups library.

[5] See [GAP08, 3.11]; this is done using a pipe.

Order 8

Together with the overhead produced by Minion – that is mostly spaces and line breaks – to output 1 843 120 128 square matrices of dimension 8 would roughly need 250 GB of disk space. In a first step to avoid this amount of data in a single file, the 3-nilpotent semigroups of order 8 are handled as a separate case. The remaining semigroups of order 8, that are the solutions of $\overline{L}_8^{-3}$ plus the zero semigroup, are partitioned depending on their diagonal. This yields 343 files containing solutions, each corresponding to a diagonal which is minimal in its equivalence class and appears in the multiplication table of a semigroup. To store the data belonging to one diagonal the same method as for semigroups of orders 2 to 7 is used, with the only difference that all diagonal entries are omitted.

Storing the 3-nilpotent semigroups of order 8 is done in a different way. To start with, not the whole tables are stored. Remember that each nilpotent semigroup has a unique generating set (Corollary 2.1.3) and that the product of two elements which are not both generators equals the zero element in a semigroup of nilpotency rank 3. Hence, if m with $2 \leq m \leq 6$ denotes the number of generators, it suffices to store the $m \times m$ tables containing all products of two generators. Then for a fixed m, the list of tables is partitioned further depending on the diagonal, as done for all other semigroups of order 8. Moreover, not all the tables are actually stored. The list of tables for a specific m and a fixed diagonal can be sorted into ranges and from each range only its length and the first table are stored. The data is collected in two separate files for each case. The first file contains the tables and is created in the same way as for the other semigroups of order 8. The lengths of the ranges are stored as a list in a second file.

To store all tables of 3-nilpotent semigroups of order 8 one would need more than 100 GB of disk space. The information actually kept uses up just under 1 GB, and is compressed with gzip to roughly 11 MB. In total, the data stored in Smallsemi related to the 1 843 973 430 multiplication tables of semigroups of order 2 to 8 takes up just under 22 MB.

4.2.2 Properties of the semigroups in the library

In addition to the semigroups themselves information about them is retained in the library. Data about those properties is kept, for which the numbers of semigroups with these properties are listed in Table 4.3. Definitions of the properties are given in Appendix B. For each property it is either stored which semigroups have it or which do not. Smallsemi itself was used to identify the semigroups with any of the properties. The identification of semigroups defining the multiplication in a near-ring relied on the SONATA [ABE+03] package. As the only exception, the self-dual semigroups were determined by adjusting the symmetry breaking method used for $\overline{L}_n$. Recall that a semigroup is self-dual if there exists an anti-isomorphism mapping the semigroup to itself, that is, an *anti-automorphism*. For a solution of the CSP $\overline{L}_n$ this translates to the condition that the set of literals forming the solution is mapped to itself under an element $g \in S_n \times C_2$ with non-trivial C_2 component. This is the case if for one of the lex-leader constraints (4.3) corresponding to an anti-isomorphism equality holds. Posting a constraint, that for at least one of these constraints equality does hold, will ensure that solutions of the modified CSP are all those solutions of $\overline{L}_n$ which lead to self-dual semigroups.

Table 4.3 Properties of semigroups up to order 8

property \ order	1	2	3	4	5	6	7	8
commutative	1	3	12	58	325	2 143	17 291	221 805
regular	1	3	9	42	206	1 352	10 168	91 073
completely regular	1	3	9	42	204	1 336	10 041	89 909
inverse	1	2	5	16	52	208	911	4 637
simple[6]	1	2	2	5	2	6	2	12
zero simple[7]	0	1	2	2	7	2	10	2
multiplication in a near-ring	1	3	5	21	9	40	14	648
self-dual	1	3	12	64	405	3 312	44 370	2 203 037
monoid	1	2	6	27	156	1 373	17 730	858 977

Definitions of the properties are given in Appendix B.

Some additional information is implicitly available due to the way the tables are stored. For orders 2 to 7 the fact that the semigroups are sorted with respect to the ordering in Equation (4.4), considering the diagonal positions first, made it possible to store efficiently which semigroups have which diagonal. For semigroups of order 8 the data is distributed on different files depending on the diagonal. So, for every semigroup the diagonal is known before the multiplication table

[6] A classification of simple semigroups is known, see [How95, Theorem 3.3.1].
[7] A classification of zero simple semigroups is known, see [How95, Theorem 3.2.3].

of the semigroup is recovered. Hence, the semigroups with a certain number of idempotents (equalling the number of fixed points of the diagonal) are quickly accessible. This applies in particular to bands.

The data related to the properties in Table 4.3 is stored in such a way that the semigroups with any combination of precomputed properties can be deduced as well. This effectively means that many more properties are precomputed. For example, a group is an inverse semigroup with one idempotent; a semilattice is a commutative band; a finite Clifford semigroup is a finite semigroup that is completely regular and inverse, and so on.

The stored information is somewhat different from what appears in the published work, in particular in the paper by Satoh et al. [SYT94]. In the latter, the numbers of semigroups with different Green's class structures are given. In Smallsemi it would not be enough to merely store the number, but the information would have to be kept for each semigroup. While this is certainly possible, storing information for every semigroup is comparable to the task of storing the semigroups themselves. As a consequence such information is not kept for all semigroups in the current version of Smallsemi. A partial exception is the number of elements in a generating set of minimal size. This information is stored for semigroups of order at most 7.

4.2.3 Usage

The functions available in Smallsemi are described in detail in its documentation, where one can find as well examples of usage. The aim here is to give a rough idea of the functionality and to show how the library might promote research.

An obvious point is that Smallsemi provides a rich set of examples of semigroups. One can analyse all or some of these semigroups simply to understand them better, possibly to come up with new conjectures or research questions. The analysis of the available semigroups of nilpotency rank $n-1$ and $n-2$ helped the author to get an intuition for those semigroups, which in the end led to their classification in Section 2.1. One specific question that is related to the studies contained in this thesis was mentioned as Question 3.2.13. Existing conjectures can be verified for orders up to 8 – or a counter example might be available.

A limitation that one has to keep in mind is that it is unfeasible to do time intensive computations for all semigroups in the library. Just to create all semigroups as objects in GAP takes around a day on a modern machine. Performing a very simple test might take the same time. To run more complicated computations might not be possible for all semigroups in a reasonable time. In such a case it is often useful that a number of properties are precomputed for the semigroups in the library. If one is interested in semigroups with any combination of precomputed properties, it is straightforward to access and test only those.

Another application is to search for minimal examples or to find additional examples of semigroups with some property. One instance of a published result which can now be verified with Smallsemi is given in [WWL81]. In this paper four semigroups of order 7 are presented as minimal examples of self-dual semigroups in which no anti-automorphism is an involution. Like all other semigroups of order at most 8, these four types of semigroups can be identified in Smallsemi. Each semigroup in the library has an ID consisting of two numbers. The first number is the order of the semigroup and the second is its position in the list of all semigroups of this order. The four semigroups from [WWL81] have – in order of appearance – IDs $(7, 646\,970), (7, 5\,693), (7, 674\,348)$, and $(7, 680\,714)$.

The identification of a semigroup returns the ID together with an isomorphism – or if no such exists an anti-isomorphism – to the equivalent semigroup in the library. This allows one to make use of fast calculations of properties for semigroups in the library and to transfer the information back to the original semigroup. While this is usually not particularly helpful for one semigroup of such small order, it becomes useful if one has a larger number of small semigroups and wants to identify equivalent ones and analyse the types of semigroups.

Finally, the fact that the self-dual semigroups in the library have been identified also allows one to efficiently work with the semigroups up to isomorphism, if needed. The result from Lemma 1.1.3 is used in the implementation providing this functionality.

5 New Enumeration Results

In the previous chapter the approach to the enumeration of semigroups using constraint satisfaction made it possible to reproduce known results easily. In this chapter the approach is improved to obtain new results, in particular the number of non-equivalent semigroups with 9 elements. Two ideas, linking to each other, allow the crucial improvements. Firstly, the enumeration problem will no longer be formulated as a single CSP, but instead as a family of CSPs. Secondly, additional mathematical knowledge is used to add implied constraints to the CSPs.

In the first section the idea of a case split leading to independent CSPs is explained on the example of having one CSP for each possible diagonal of the multiplication table of a semigroup. The example arises from the idea introduced by Plemmons [Ple67]. The chapter continues utilising the idea of a case split for the enumeration of bands in the second chapter. Detailed structural information about bands is employed. Non-equivalent bands up to order 10 are constructed. Their number of order 9 is 618 111, and there are 7 033 090 of order 10. The results up to that point are sufficient to report in the third section the number of semigroups of order 9 up to equivalence to be 52 989 400 714 478. Some further refinements follow, which aim in particular to reduce the space requirements of the computations.

Section 5.4 contains results first published by the author and Kelsey in [DK09]. Another application of the methods from the first section led to the numbers of monoids with at most 10 elements. For orders up to 8 they are included in Smallsemi [DM10], see Table 4.3. Up to equivalence there are 1 844 075 697 monoids of order 9 and 52 991 253 973 742 of order 10.

In Section 5.5 the automorphism groups of all semigroups of order at most 9 are determined using CSPs. Only the automorphism groups of semigroups up to order 7 were known before [ABMN09]. This shows that constraint satisfaction can even be employed as a tool in the analysis of algebraic objects.

The chapter – and the thesis – close with an outlook on possible future applications of the presented techniques.

5.1 A Family of CSPs

The possibilities to enhance the CSP L_n are restricted simply by the fact that not much can be said about the multiplication table of a semigroup in general without knowing any of the entries. The original idea by Plemmons [Ple67] to make the search for multiplication tables of semigroups more efficient, was to do separate computations depending on the diagonal. To consider diagonal positions first in the search, as done in the previous chapter, was the adaptation of this idea used by Satoh et al. [SYT94]. To actually do separate computations more than one CSP has to be formulated.

CSP 5.1.1 Given a function $f : [n] \to [n]$ (corresponding to the diagonal of a multiplication table, see Chapter 3) define a CSP $L_f = (V_n, D_n, C_f)$ based on L_n (CSP 4.1.3) by adding the constraints

$$T_{i,i} = f(i) \text{ for all } i \in [n] \tag{5.1}$$

to C_n to obtain C_f.

The solutions to L_f are all multiplication tables defining a semigroup where the square of the element i is given by $f(i)$. In other words, the entries on the diagonal of the multiplication table are given. For a set $\mathcal{F}$ of functions from $[n]$ to $[n]$ denote by $\mathcal{L}_{\mathcal{F}}$ the family of CSPs $\{L_f \mid f \in \mathcal{F}\}$.

Let $\mathcal{F}_n$ denote the set of all functions from $[n]$ to $[n]$. Then the CSPs in $\mathcal{L}_{\mathcal{F}_n}$ have together the same solutions as L_n. It was already mentioned in Chapter 3 that it suffices to consider a subset of all functions if one only wants one semigroup of each equivalence class. As the split of one CSP into a whole family will occur repeatedly in this chapter, which set of diagonals to use shall be answered in a more general setting.

Lemma 5.1.2 *Let $\mathcal{L} = \{L_x \mid x \in X\}$ be a family of CSPs with disjoint solution sets, and let $\mathcal{T}$ be a superset of all solutions. Let $\phi : \mathcal{T} \times G \to \mathcal{T}, (T, g) \mapsto T^g$ be* *an action of a group G mapping solutions to solutions. Further let $\psi : \mathcal{T} \to X$ be a surjective function.*

If, for all $x \in X$ and for every solution T of L_x, T is a solution of $L_{\psi(T)}$, and if ϕ^ψ is an induced action of G on X (that is, $x^g = \psi(T^g)$ for $x = \psi(T)$ is well-defined), then the following statements hold.

(i) *Let $Y \subseteq X$ contain at least one element of every orbit from X under the induced action ϕ^ψ. Then the solutions of $\{L_y \mid y \in Y\}$ contain at least one element from every orbit of solutions under the action of ϕ.*

(ii) *Let $S \in L_x$ and $T \in L_y$. If x is not equivalent to y, then S is not equivalent to T.*

(iii) *Let $T \in L_x$. Then the set of solutions of L_x equivalent to T is the orbit of T under the stabiliser of x in G.*

Proof: (i): Let T be a solution of one of the instances in $\mathcal{L}$. By assumption T is a solution of $L_{\psi(T)}$ and there exists a $y \in Y$ equivalent to $\psi(T)$, that is $\psi(T)^g = y$ for some $g \in G$. As $\psi(T)^g = \psi(T^g)$, it follows that T^g is a solution of $L_y = L_{\psi(T^g)}$.

(ii): To show the contraposition of the second statement let T be equivalent to S. Thus $T^g = S$ for some $g \in G$. Note that $x = \psi(S)$ and $y = \psi(T)$ as the solution sets of different instances are disjoint. Hence, $x = \psi(S) = \psi(T^g) = \psi(T)^g = y^g$, showing that x is equivalent to y.

(iii): Let $g \in G$ be arbitrary. Then T^g is a solution of $L_{\psi(T^g)}$. Since the CSPs in $\mathcal{L}$ have disjoint solution sets, T^g is a solution of L_x if and only if $\psi(T^g) = x^g = x$. Hence T^g is a solution of L_x if and only if g lies in the stabiliser of x in G. □

Choosing $\mathcal{L} = \mathcal{L}_{\mathcal{F}_n}$, $\mathcal{T} = \Omega_n$ (the set of all multiplication tables on $[n]$), $G = S_n \times C_2$, and ψ as the mapping sending multiplication tables to the function corresponding to their diagonal, satisfies the conditions in Lemma 5.1.2. The induced action of $S_n \times C_2$ on diagonals, and hence on functions from $[n]$ to $[n]$, was described in Section 3.1 together with an algorithm to obtain a set of diagonals up to equivalence. If $\overline{\mathcal{F}}_n$ denotes such a set of non-equivalent diagonals, respectively functions, then each type of semigroup appears as solution of $\mathcal{L}_{\overline{\mathcal{F}}_n}$ due to Lemma 5.1.2(i). Moreover, different CSPs in $\mathcal{L}_{\overline{\mathcal{F}}_n}$ have pairwise non-equivalent solutions due to Lemma 5.1.2(ii). This allows one to search independently in different

CSPs for non-equivalent solutions, and the solutions of $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_n} = \{\overline{L}_f \mid f \in \overline{\mathcal{F}}_n\}$ will form a set of semigroups on $[n]$ up to equivalence. The solutions of L_f form orbits under the stabiliser of f in $S_n \times C_2$ according to Lemma 5.1.2(iii). Hence, adding one lex-leader constraint (4.3) for every element in $\mathrm{Stab}_{S_n \times C_2}(f)$ to L_f yields $\overline{L}_f$ following the considerations in Section 4.1.2.

With n getting up to 9 or even 10, calculating the stabiliser in $S_n \times C_2$ of a function f corresponding to a diagonal directly under the induced action, starts to become an efficiency issue for setting up the input files. This can be avoided by reformulating the action to a pointwise action on sets. Then sophisticated algorithms, in particular partition backtrack [Leo91], are available performing the stabiliser calculation far more efficiently. The reformulation was in principal already introduced when explaining lex-leader constraints in Subsection 4.1.2. Every element $g \in S_n \times C_2$ induces a bijection of the literals of the CSP L_f. Take the set of literals $\chi_f = \{(T_{i,i} = f(i)) \mid 1 \leq i \leq n\}$ corresponding to the fixed diagonal entries. Then g is in the stabiliser of f if and only if $\chi_f^g = \chi_f$. It is not a coincidence that the stabiliser of f in $S_n \times C_2$ equals the stabiliser of a set of literals; note the following result.

Lemma 5.1.3 *Let $L = (V_n, D_n, C)$ be a CSP with non-empty solution set, in which each set of equivalent solutions forms an orbit under $G \leq S_n \times C_2$. If there exists a subset χ of all literals χ_L such that the solutions of L are the subsets of χ that are full assignments, then each set of equivalent solutions forms an orbit under the setwise stabiliser of χ in $S_n \times C_2$.*

Proof: The statement is shown in two steps. First, $G \leq \mathrm{Stab}_{S_n \times C_2}(\chi)$ is proven. Let $g \in G$ and denote the set of solutions of L by $\mathcal{T}$. Note that χ equals the union of all solutions. Then

$$\chi^g = \left(\bigcup_{T \in \mathcal{T}} T\right)^g = \bigcup_{T \in \mathcal{T}} T^g = \bigcup_{T \in \mathcal{T}} T = \chi$$

and hence $g \in \mathrm{Stab}_{S_n \times C_2}(\chi)$.

It remains to be shown that the image of every solution T of L under an element g in $\mathrm{Stab}_{S_n \times C_2}(\chi)$ is a solution equivalent to T. This follows immediately from $T^g \subseteq \chi$. $\square$

Table 5.1 Enumeration of non-equivalent semigroups on $[n]$ using a family of CSPs

n	1	2	3	4	5	6	7	8
$\overline{\mathcal{L}}_{\overline{\mathcal{F}}_n}$, solutions	1	4	18	126	1 160	15 973	836 021	1 843 120 128
–, instances	1	2	5	13	34	60	243	660
–, setup time	ϵ	ϵ	ϵ	ϵ	1 s	9 s	68 s	689 s
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	4 s	35 s	14 015 s
$\overline{\mathcal{L}}^{-3}_{\overline{\mathcal{F}}_n}$, solutions	0	3	16	117	1 075	13 312	226 223	11 433 105
–, instances	1	2	5	13	34	60	243	660
–, setup time	ϵ	ϵ	ϵ	ϵ	1 s	9 s	82 s	667 s
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	4 s	31 s	563 s

The times are rounded to seconds. They were obtained using the 64-bit executable of Minion, version 0.9, on a machine with 2.66 GHz Intel X-5430 processor and 16 GB RAM. The setup was done with GAP. The symbol ϵ stands for a time less than 0.5 s.

The previous lemma does not directly apply to the CSPs in $\mathcal{L}_{\mathcal{F}_n}$, because of the associativity constraint. If one neglects this constraint, such that the solutions are all magmas fulfilling the remaining constraints, then the assumptions of Lemma 5.1.3 are satisfied. Any full instantiation for which the values on the diagonal are in χ_f is a solution for $f \in \mathcal{F}_n$, and equivalent solutions form orbits under the stabiliser of the literals in $S_n \times C_2$. Adding the associativity constraint back in does not change this fact, since associativity is invariant under isomorphism and anti-isomorphism.

As before a CSP L_f^{-3} is defined by adding constraint (4.2) to L_f, ruling out 3-nilpotent solutions. Adding this constraint is only necessary for functions f fulfilling the criterion in Lemma 3.2.5 for 3-nilpotent semigroups, as otherwise L_f does not have 3-nilpotent solutions. The family of instances $\{L_f^{-3} \mid f \in \overline{\mathcal{F}}_n\}$ is denoted by $\mathcal{L}^{-3}_{\overline{\mathcal{F}}_n}$.

The results of solving, for $1 \leq n \leq 8$, the instances constructed from $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_n}$ and $\overline{\mathcal{L}}^{-3}_{\overline{\mathcal{F}}_n}$ with Minion [GJM06] are given in Table 5.1. The numbers of semigroups coincide with those from Table 4.2, while the runtimes for $n = 8$ are a magnitude smaller.

That the diagonal entries of the multiplication table are known in L_f has further consequences. The most obvious one, following the analysis in Section 3.2, is that the CSP L_f has no solutions if the diagonal corresponding to f does not allow the table to be completed such that an associative multiplication is defined. Another point is that some of the constraints enforcing associativity are trivially satisfied or simplify, since the product of two elements is known whenever both factors are the same. Moreover, for $i \in [n]$ all powers of the form $i^{2^k}, k \in \mathbb{N}$ can be computed from the diagonal (and no new values occur for $k > n$). If j equals any such power, then $ij = ji$. This leads to the implied constraint

$$T_{i,j} = T_{j,i} \text{ for all } i, j \in [n] \text{ with } f^k(i) = j \text{ for some } k \in [n] \tag{5.2}$$

for L_f. More implied constraints can be added to L_f in the special case where f has exactly one fixed point and the digraph corresponding to f contains cycles. According to Lemma 3.2.7 elements in $[n]$ labelling the vertices of the cycle lie in a common subgroup with the unique idempotent as identity. The computational effect of all considerations in this paragraph on $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_n}$ is very minor. None of them relates to the bottleneck of the computation which is the CSP L_{id_n}, where id_n denotes the identity function on $[n]$. This instance is dealt with in the following section.

In [Jür78] Jürgensen claims that a case split depending on the number of idempotents in the semigroup is nearly as powerful as the case split depending on diagonals. That the additional information available from the diagonal has no considerable influence on the speed of the computation might be mistaken to support this claim. On the contrary, the fact that implied constraints do not help the computation, indicates that the propagation of (4.1), enforcing associativity, works well already. It does not imply that the information from the diagonal is superfluous regarding the efficiency. For a rigorous verification of Jürgensen's claim a case split on the number of idempotents was implemented. For a non-empty subset U of $[n]$ let L_U denote a CSP based on L_n with the additional constraints

$$\begin{aligned} T_{i,i} &= i \quad \text{if} \quad i \in U \\ T_{i,i} &\neq i \quad \text{if} \quad i \in [n] \setminus U. \end{aligned}$$

The group used to add lex-leader constraints (4.3) to L_U to obtain $\overline{L}_U$ is the direct product $(S_U \times S_{[n]\setminus U}) \times C_2$. Then $\overline{\mathcal{L}}_{\overline{\mathcal{U}}} = \{\overline{L}_U \mid U = [m], 1 \leq m \leq n\}$ is a family of CSPs with all semigroups up to equivalence as solutions. Note that $L_{[n]} = L_{\mathrm{id}_n}$.

Table 5.2 Case split on the number of idempotents

n	1	2	3	4	5	6	7	8
$\overline{\mathcal{L}}_{\overline{\mathcal{U}}}$, solutions	1	4	18	126	1 160	15 973	836 021	1 843 120 128
–, instances	1	2	3	4	5	6	7	8
–, setup time	ϵ	ϵ	ϵ	ϵ	ϵ	3 s	42 s	525 s
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	1 s	33 s	519 542 s
$\overline{\mathcal{L}}_{\overline{\mathcal{U}}}^{-3}$, solutions	0	3	16	117	1 075	13 312	226 223	11 433 105
–, instances	1	2	3	4	5	6	7	8
–, setup time	ϵ	ϵ	ϵ	ϵ	ϵ	3 s	40 s	518 s
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	1 s	13 s	819 s

The times are rounded to seconds. They were obtained using the 64-bit executable of Minion in version 0.9 on a machine with 2.66 GHz Intel X-5430 processor and 16 GB RAM. The setup was done with GAP. The symbol ϵ stands for a time less than 0.5 s.

The results for Jürgensen's approach are given in Table 5.2. The runtime for $n = 8$ is only marginally slower than for the family $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_8}^{-3}$. The bottleneck of both computations regarding memory usage is the same CSP, that is $L_{[n]} = L_{\mathrm{id}_n}$.

One aspect, that typically influences the runtime, when some kind of equivalence is involved, is the choice of representatives. For two equivalent functions f and h the time to solve L_f and L_h can indeed vary considerably. Some testing indicates that the representative returned by the implementation based on Algorithm 1 is a good candidate. Again, this is not relevant for the bottleneck of the computation, L_{id_n}, since the identity function forms its own equivalence class. Satoh et al. state in [SYT94] that their program gives – after filling the positions on the diagonal – second priority to the positions in the multiplication table that are in the row or column of an idempotent. For the approach using constraint satisfaction, tests on some diagonals do not support this preference for the ordering of table positions. In particular, for diagonals with a cycle in the digraph corresponding to the diagonal it seems most beneficial to first fill the positions in the row and column of elements from the cycle.

This concludes the reproduction of and comparison with former results on the enumeration of semigroups. In the next section a far more efficient way to solve the bottleneck of the family of CSPs $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_n}$ is presented, which subsequently results in the finding of the number of semigroups of order 9 up to equivalence.

5.2 Bands

One might think that subdividing a CSP into a family of independent instances is a good way to parallelise the computation. This is not true for the family $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_n}$. The biggest difficulties lie in a small number of instances. This is mainly influenced by two factors: the number of solutions a particular instance $\overline{L}_f$ has, and how big the stabiliser in $S_n \times C_2$ of the diagonal corresponding to f is. The former varies a lot from case to case with most functions leading to no semigroup as discussed in Section 3.2. The more severe problem at this point is the latter, which is best demonstrated on the extreme example where f is the identity function id_n on $[n]$. Every element in $S_n \times C_2$ stabilises the diagonal corresponding to id_n. Hence $2n!$ lex-leader constraints are added for the symmetry breaking. As one effect, solving the instance $\overline{L}_{\mathrm{id}_n}$ takes more memory than all other instances in $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_n}$. For $n = 8$ the computation uses roughly 8 GB RAM. Since the memory usage is linear in the number and length of the lex-leader constraints (4.3), the computation is not possible for $n = 9$ on the available machine with 16 GB of memory.

That the instance $\overline{L}_{\mathrm{id}_n}$ is completely independent of all other instances in $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_n}$ allows one to use additional mathematical knowledge which is available just for this instance. The solutions of $\overline{L}_{\mathrm{id}_n}$ correspond to the bands on $[n]$ up to equivalence. The structure of bands is well understood. The building blocks for bands are two types of bands with additional properties. These types are introduced in the next definition.

Definition 5.2.1 Let B be a band.

(i) If B is commutative, then B is a *semilattice*.

(ii) If $sts = s$ holds for all $s, t \in B$, then B is a *rectangular band*.

The operation in a semilattice will usually be denoted by $\wedge$. To obtain the semilattices on $[n]$ up to equivalence define a CSP $\overline{SL}_n$ based on $\overline{L}_{\mathrm{id}_n}$ by adding the constraint

$$T_{i,j} = T_{j,i} \text{ for all } i, j \in [n]$$

to enforce commutativity. In a semilattice B, saying '*s is smaller than t*' if $s \wedge t = s$ for $s, t \in B$ yields a partial order on B. Using the partial order in combination with the minimality of tables that are solutions of $\overline{SL}_n$ leads to an additional property of solutions, stated in the following lemma.

Lemma 5.2.2 *Let B be a semilattice (with operation $\wedge$) corresponding to a solution of $\overline{SL}_n$. Then the inequality $i \wedge j \leq \min\{i,j\}$ holds for all $i,j \in [n]$.*

Proof: Let T be a table defining a semilattice on $[n]$ for which there exist $k,l \in [n]$ such that $k \wedge l > \min\{k,l\}$. Without loss of generality let (k,l) be the first position (with respect to the order of table positions (4.4) used for the search) for which this inequality holds. It will be shown that $T^{(k\,k\wedge l)} < T$ and hence T is not a solution of $\overline{SL}_n$. Clearly, $k < l$ by commutativity, and by assumption $(T^{(k\,k\wedge l)})_{k,l} = (T_{k\wedge l,l})^{(k\,k\wedge l)} = k < k \wedge l = T_{k,l}$. It suffices to show that for every position (i,j) that comes earlier in the order of positions (4.4) the inequality $(T^{(k\,k\wedge l)})_{i,j} \leq T_{i,j}$ holds. Positions of the form (i,i) do not have to be considered since the diagonal corresponds in any case to the constant function id_n.

Case 1: consider a position (i,j) with $i,j \notin \{k, k\wedge l\}$, and $i \neq j$. Then $(T^{(k\,k\wedge l)})_{i,j} = (T_{i,j})^{(k\,k\wedge l)}$. If for some position $(T_{i,j})^{(k\,k\wedge l)} > T_{i,j}$ holds, then it follows $T_{i,j} = k$. Hence, $k = i \wedge j$ which implies $i \wedge k = i \wedge i \wedge j = i \wedge j = k$. Due to the minimality of (k,l) it follows $k \leq \min\{i,k\}$ and in particular $k \leq i$, in fact $k < i$. Consequently (i,j) comes later than (k,l) in the order of positions.

Case 2: for $i < k$ consider the two positions (i,k) and $(i, k\wedge l)$. Neither of the entries at the two positions in T equal k or $k \wedge l$ as this would contradict minimality of (k,l). Then $(T^{(k\,k\wedge l)})_{i,k} = (T_{i,k\wedge l})^{(k\,k\wedge l)} = T_{i,k\wedge l}$ and $(T^{(k\,k\wedge l)})_{i,k\wedge l} = (T_{i,k})^{(k\,k\wedge l)} = T_{i,k}$. As (i,k) comes earlier than $(i,k\wedge l)$ in the order of positions (4.4) it suffices to show $p = (T^{(k\,k\wedge l)})_{i,k} = T_{i,k\wedge l} \leq T_{i,k} = q$. From the minimality of (k,l) it follows that $p,q \leq i$. Hence $p \wedge q = (i \wedge k \wedge l) \wedge (i \wedge k) = i \wedge k \wedge l = p$ which implies $p \leq \min\{p,q\}$ and consequently $p \leq q$.

Case 3: for $j < k$ consider the two positions (k,j) and $(k\wedge l, j)$. Analogously to Case 2 one shows that $(T^{(k\,k\wedge l)})_{k,j} = T_{k\wedge l,j} \leq T_{k,j}$, which completes the proof that $T^{(k\,k\wedge l)}$ is smaller than T. $\square$

From the result in the previous lemma it follows that

$$T_{i,j} \leq \min\{i,j\} \tag{5.3}$$

is an implied constraint for $\overline{SL}_n$. Hence it can be added to the CSP without changing the solution set. As constraint (5.3) causes considerable domain restrictions for many variables, adding it reduces the search space drastically. The results and

Table 5.3 Enumeration of non-equivalent bands on $[n]$ using a family of CSPs

n	1	2	3	4	5	6	7	8	9	10
$\overline{L}_{\mathrm{id}_n}$, solutions	1	2	6	26	135	875	6 749	60 601	*618 111*	*7 033 090*
–, setup time	ϵ	ϵ	ϵ	ϵ	ϵ	2 s	30 s	361 s	$\diamond$	$\diamond$
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	3 s	171 s	$\diamond$	$\diamond$
$\overline{SL}_n$, solutions	1	1	2	5	15	53	222	1 078	5 994	37 622
–, setup time	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	4 s	54 s	711 s
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	5 s	150 s
$\overline{\mathcal{L}}_{\overline{\mathcal{B}}_n^*}$, solutions	0	0	3	19	119	820	6 526	59 521	612 115	6 995 466
–, instances	0	0	2	9	39	165	784	4 181	25 037	167 059
–, setup time	–	–	ϵ	ϵ	1 s	10 s	85 s	985 s	12 916 s	144 314 s
–, solve time	–	–	ϵ	ϵ	ϵ	1 s	11 s	90 s	791 s	9 322 s
$\overline{\mathcal{L}}_{\overline{\mathcal{R}}_n}$, solutions	1	2	6	26	135	875	6 749	60 601	618 111	*7 033 090*
–, instances	1	2	3	5	6	8	9	11	13	*15*
–, setup time	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	5 s	56 s	676 s	$\diamond$
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	16 s	819 s	$\diamond$

The times are rounded to seconds. They were obtained using the 64-bit executable of Minion version 0.9 on a machine with 2.66 GHz Intel X-5430 processor and 16 GB RAM. The setup was done with GAP. The symbol ϵ stands for a time less than 0.5 s. The symbol $\diamond$ indicates insufficient memory.

runtimes of $\overline{SL}_n$ for $1 \leq n \leq 10$ are given in Table 5.3. It is well-known that semilattices on $[n]$ are in one to one correspondence with lattices on $[n+1]$.[1] The numbers of semilattices on $[n]$ are therefore known for n up to 17 from [HR02], in which lattices with up to 18 elements are counted. Note that [HR02] is another example for the application of an 'orderly algorithm', the constructive enumeration technique which was introduced by Read [Rea78] and mentioned in the final paragraph of Section 3.1.

The second type of band introduced in Definition 5.2.1 does not need to be searched for. Rectangular bands are isomorphic to semigroups on a Cartesian product $I \times \Lambda$ with multiplication defined by $(i, \lambda)(j, \mu) = (i, \mu)$, and each such multiplication defines a rectangular band. Two rectangular bands $I_1 \times \Lambda_1$ and $I_2 \times \Lambda_2$ are isomorphic if and only if $|I_1| = |I_2|$ and $|\Lambda_1| = |\Lambda_2|$, and they are anti-isomorphic if and only if $|I_1| = |\Lambda_2|$ and $|\Lambda_1| = |I_2|$. Hence, the number of

[1]The lattice corresponding to a semilattice B is obtained by adding a new identity element and defining a second operation $\vee$ by $s \vee t = \bigwedge\{x \in B^1 \mid x \wedge s = s \;\&\; x \wedge t = t\}$ for all $s, t \in B^1$.

rectangular bands on $[n]$ up to isomorphism equals the number of divisors of n, and the number of rectangular bands on $[n]$ up to equivalence equals the number of divisors of n less or equal $\sqrt{n}$.

Knowing the building blocks, the statement about the structure of bands is given next.

Theorem 5.2.3 *Let B be a band and let $\{R_\alpha \mid \alpha \in Y\}$ be the set of $\mathcal{D}$-classes of B. Then the following statements hold:*

(i) *for all $\alpha \in Y$ the $\mathcal{D}$-class R_α forms a subsemigroup that is a rectangular band;*

(ii) *for all $\alpha, \beta \in Y$ the set $R_\alpha R_\beta = \{ab \mid a \in R_\alpha, b \in R_\beta\}$ is contained inside a $\mathcal{D}$-class of B;*

(iii) *the index set Y forms a semilattice under the operation $\wedge$ defined through $R_\alpha R_\beta \subseteq R_{\alpha\wedge\beta}$.*

The previous theorem is a special case of a more general result stating that every completely regular semigroup is a semilattice of completely simple semigroups [How95, Theorem 4.1.3]. The statement of Theorem 5.2.3 is given in [How95, Theorem 4.4.1] in a compact form and a proof – as well for the generalisation to regular semigroups – can be found in the same reference. Moreover, there is a complete characterisation for bands available [How95, Theorem 4.4.5]. This characterisation defines the multiplication in a band via mappings between the rectangular bands fulfilling certain conditions to guarantee associativity. Since, using a CSP to search for bands, the conditions for the mappings are not easier to check than associativity itself, only the information from Theorem 5.2.3 will be used for the search.

CSP 5.2.4 Let Y be a semilattice, and let $\{R_\alpha \mid \alpha \in Y\}$ be a set of disjoint rectangular bands with $\bigcup_{\alpha\in Y} R_\alpha = [n]$. Then define, based on L_n from CSP 4.1.3, a CSP $L_{\{R_\alpha|\alpha\in Y\}} = (V_n, D_n, C_{\{R_\alpha|\alpha\in Y\}})$ by adding the constraints

$$T_{i,j} = ij \text{ for all } i, j \in R_\alpha \text{ for all } \alpha \in Y, \tag{5.4}$$

$$T_{i,j} \in R_{\alpha\wedge\beta} \text{ for all } i \in R_\alpha, j \in R_\beta. \tag{5.5}$$

Each solution of CSP 5.2.4 is obviously a table defining a band. Moreover, every such CSP has at least one solution, as will be explained in the following remark. One word about the notation: if $\{R_\alpha \mid \alpha \in Y\}$ are the $\mathcal{D}$-classes of a band where Y is a semilattice, then this shall implicitly mean that the operation on Y arose from the multiplication in the band as described in Lemma 5.2.3.

Remark 5.2.5 Given a set of rectangular bands, $\{(R_\alpha, *_\alpha) \mid \alpha \in Y\}$, forming a semilattice, there exists a band having this $\mathcal{D}$-class structure. To define a multiplication choose fixed elements $s_\alpha \in R_\alpha$ for all $\alpha \in Y$ and then define the product ab for $a \in R_\alpha$ and $b \in R_\beta$ by

$$ab = \begin{cases} a *_\alpha b & \text{if } \alpha = \beta \\ a *_\alpha s_\alpha & \text{if } \alpha \wedge \beta = \alpha \\ s_\beta *_\beta b & \text{if } \alpha \wedge \beta = \beta \\ s_{\alpha \wedge \beta} & \text{otherwise.} \end{cases}$$

Following on from the comments right after Theorem 5.2.3, the statement in the previous remark generalises as well to the case of semilattices of completely simple semigroups. A completely regular semigroup arising from such a construction is a *strong semilattice of completely simple semigroups*. The given construction is an adaptation of the explanation at the end of [How95, Section 4.1]. Note that not every band arises from such a construction. On the other hand, for the degenerate case of a $\mathcal{D}$-class structure $\{R_\alpha \mid \alpha \in Y\}$ with $|Y| = n$ or $|Y| = 1$ there is exactly one band with the given structure. In the former case it is the semilattice Y itself, and in the latter case it is the unique rectangular band $R_\alpha, \alpha \in Y$ in the set. (More generally, every $\mathcal{D}$-class structure, in which each rectangular band R_α contains only one element if α is not maximal in Y, allows exactly one multiplication.)

To make sure the CSPs from 5.2.4 are useful for a case split to find all bands, more precise information is needed.

Lemma 5.2.6 *The solutions of $L_{\{R_\alpha \mid \alpha \in Y\}}$ are all multiplication tables defining bands with $\{R_\alpha \mid \alpha \in Y\}$ as $\mathcal{D}$-classes.*

Proof: It shall first be shown that every solution leads to a band with $\mathcal{D}$-classes $\{R_\alpha \mid \alpha \in Y\}$. Let T be a solution table. Constraint (5.4) fixes parts of the multiplication table according to the multiplication in the rectangular bands $R_\alpha, \alpha \in Y$.

For two elements $s, t \in R_\alpha$ both $sts = s$ and $tst = t$ hold, due to Definition 5.2.1(ii). Hence, the elements in R_α are contained in the same $\mathcal{D}$-class. For $s \in R_\alpha$ and $t \in R_\beta$ assume $s\mathcal{D}t$. Thus there exist $u \in R_\gamma$ for some $\gamma \in Y$, and $v \in R_\delta$ for some $\delta \in Y$ such that $t = usv$. From constraint (5.5) it follows that $\beta = \gamma \wedge \alpha \wedge \delta$ and therefore that $\alpha \wedge \beta = \beta$. Exchanging s and t in this argument one obtains $\alpha \wedge \beta = \alpha$, which yields $\beta = \alpha$. Hence, two elements are $\mathcal{D}$-related if and only if they lie in the same rectangular band.

Let now T be the multiplication table of a band having $\mathcal{D}$-classes $\{R_\alpha \mid \alpha \in Y\}$. Multiplication inside a $\mathcal{D}$-class is given by the respective rectangular band and thus T complies with constraint (5.4). Products from distinct $\mathcal{D}$-classes respect the structure of the semilattice Y by Theorem 5.2.3, meaning that $R_\alpha R_\beta \subseteq R_{\alpha\wedge\beta}$. Thus the entries in T fulfil constraint (5.5) and – since it defines an associative multiplication table – T is a solution of $L_{\{R_\alpha|\alpha\in Y\}}$. □

Unfortunately introducing indices for the rectangular bands means that two different sets $\{D_\beta \mid \beta \in X\}$ and $\{R_\alpha \mid \alpha \in Y\}$ might describe the same semilattice of rectangular bands. The CSPs $L_{\{D_\beta|\beta\in X\}}$ and $L_{\{R_\alpha|\alpha\in Y\}}$ are identical if and only if there exists an isomorphism $\sigma : X \to Y$ such that $D_\beta = R_\alpha$ whenever $\sigma(\beta) = \alpha$. To avoid this ambiguity, the first step is to consider semilattices in the following up to isomorphism, that is, each semilattice shall be a solution of $\overline{SL}_n$. Still, σ might induce an automorphism. One can easily define a canonical indexing by choosing the indices in $\{R_\alpha \mid \alpha \in Y\}$ such that the inequality

$$\left(\min R_1, \min R_2, \dots, \min R_{|Y|}\right) \leq_{lex} \left(\min R_{1^\pi}, \min R_{2^\pi}, \dots, \min R_{|Y|^\pi}\right) \tag{5.6}$$

holds for all automorphisms π of Y. On the other hand, in situations where the specific indexing is not important, it can be more useful to simply state that two sets $\{D_\beta \mid \beta \in Y\}$ and $\{R_\alpha \mid \alpha \in Y\}$ are identified if they define the same CSP.

From Remark 5.2.5 and Lemma 5.2.6 it follows that all instances $L_{\{R_\alpha|\alpha\in Y\}}$ have at least one solution. Apart from this fact the situation is now similar to that in Section 5.1, when trying to find all semigroups on $[n]$ up to equivalence using the family of instances $\mathcal{L}_{\mathcal{F}_n}$. It will be shown that all assumptions for Lemma 5.1.2 are fulfilled. The set $\mathcal{B}_n$ shall contain all possible $\mathcal{D}$-class structures of bands on $[n]$. Then the solution sets of different instances in the family $\mathcal{L}_{\mathcal{B}_n} = \left\{L_{\{R_\alpha|\alpha\in Y\}} \mid \{R_\alpha \mid \alpha \in Y\} \in \mathcal{B}_n\right\}$ are disjoint due to Lemma 5.2.6. The group G is again $S_n \times C_2$ acting this time on the set of all multiplication tables on $[n]$ that define bands, which is identical with the set of all solutions. Finally, let ψ be the function sending the multiplication table of a band to the $\mathcal{D}$-class structure of the band. Then every solution T of an instance in $\mathcal{L}_{\mathcal{B}_n}$ is a solution of $L_{\psi(T)}$ by Lemma 5.2.6. Furthermore, the induced action of $S_n \times C_2$ on the sets $\{R_\alpha \mid \alpha \in Y\}$ via $\{R_\alpha \mid \alpha \in Y\}^g = \{R_\alpha^g \mid \alpha \in Y\}$ for $g \in S_n \times C_2$ is well-defined, because if the $\mathcal{D}$-class structures of two bands coincide, so will the $\mathcal{D}$-class structures of their images under the same isomorphism or anti-isomorphism.

To actually find representatives for every $\mathcal{D}$-class structure under the induced action of $S_n \times C_2$ is computationally non-trivial. The two main problems are the size of the set acted upon and that the convention from (5.6) has to be taken into account. As when constructing diagonals in Section 3.1 the orbit calculations can be simplified. Algorithm 2 uses invariants of the sets $\{R_\alpha \mid \alpha \in Y\}$ to split the set of all $\mathcal{D}$-class structures into unions of orbits, which are considered independently. The sizes of the rectangular bands (as multiset) is one of the invariants used. A set would then be canonical if first

$$\big(|R_1|, |R_2|, \ldots, |R_{|Y|}|\big) \leq_{lex} \big(|R_{1^\pi}|, |R_{2^\pi}|, \ldots, |R_{|Y|^\pi}|\big) \tag{5.7}$$

holds for all automorphisms π of Y, and of all such configurations inequality (5.6) holds for the canonical one.

Three procedures are assumed to exist as prerequisite for Algorithm 2.

Partitions takes two positive integers N and K as input and outputs all partitions of N with K summands of positive integers.

Orbits takes a group and a set as input; the set being closed under the action of the group. It outputs the orbits on the set under the action of the group.

Stabiliser takes as input a group and an element from a set the group acts on. It outputs the largest subgroup of the group acting trivially on the element.

For all three procedures existing implementations in GAP are used.

Lemma 5.2.7 *Algorithm 2 is correct.*

Proof: It needs to be shown that no two $\mathcal{D}$-class structures in the output are equivalent, and that for every $\mathcal{D}$-class structure an equivalent one appears in D.

Algorithm 2

Construct non-equivalent $\mathcal{D}$-class structures for bands on $[N]$ with K $\mathcal{D}$-classes

Require: $K \leq N$

1: $D \leftarrow [\,]$ {initialise output as empty list}
2: **for all** $L \in \overline{SL}_K$ **do**
3:   **for all** $p_1 + p_2 + \cdots + p_K \in \textsc{Partitions}(N, K)$ **do** {the partition specifies the sizes of the rectangular bands}
4:     $G \leftarrow \textsc{Stabiliser}(S_K, L)$ {automorphism group of the semilattice L}
5:     $\mathcal{A} \leftarrow \{(p_{1^\pi}, p_{2^\pi}, \ldots, p_{K^\pi}) \mid \pi \in S_K\}$ {all arrangements of $(p_1, p_2, \ldots, p_K)$}
6:     **for all** $\mathcal{O}_1 \in \textsc{Orbits}(G, \mathcal{A})$ **do**
7:       $(q_1, q_2, \ldots, q_K) \leftarrow$ representative of $\mathcal{O}_1$
8:       **for all** $i \in \{1, 2, \ldots, K\}$ **do**
9:         $\mathcal{R}_i \leftarrow$ non-isomorphic rect. bands on $\left\{\sum_{j=1}^{i-1} q_j + 1, \ldots, \sum_{j=1}^{i} q_j\right\}$
10:       **end for**
11:       $\mathcal{C} \leftarrow$ the K-fold Cartesian product of $\{\mathcal{R}_i \mid 1 \leq i \leq K\}$
12:       $H \leftarrow \textsc{Stabiliser}(G, (q_1, q_2, \ldots, q_K))$
13:       **for all** $\mathcal{O}_2 \in \textsc{Orbits}(H \times C_2, \mathcal{C})$ **do**
14:         $(R_1, R_2, \ldots, R_K) \leftarrow$ representative of $\mathcal{O}_2$
15:         add pair $(L, (R_1, R_2, \ldots, R_K))$ to list D
16:       **end for**
17:     **end for**
18:   **end for**
19: **end for**
20: **return** D

Two sets $\{R_\alpha \mid \alpha \in Y\}$ and $\{D_\beta \mid \beta \in X\}$ are not isomorphic if Y is not isomorphic to X or if the sizes of the rectangular bands do not match. Hence, sets constructed in lines 4 to 17 of the algorithm may only be equivalent if they are constructed in the same run. On the other hand, there is a run for all possible sizes and isomorphism types of semilattices.

The representative structure under automorphisms of the semilattice can be chosen in two steps as described in the paragraph before the lemma. A representative vector of the sizes of bands under the automorphisms of the semilattice is chosen in line 6. No two structures constructed in different runs through this loop can be equivalent. Moreover, for every structure $\{R_\alpha \mid \alpha \in Y\}$ with the vector $\big(|R_{1^\pi}|, |R_{2^\pi}|, \ldots, |R_{|Y|^\pi}|\big)$ of sizes of the rectangular bands, there is a run with an equivalent vector of sizes under an automorphism of the semilattice Y.

In the second step only automorphisms of the semilattice stabilising the vector of sizes are considered. This subgroup is computed in line 12. A representative of an orbit of tuples of rectangular bands under automorphisms and anti-automorphisms of the semilattice fixing the vector of sizes is chosen in line 13. Hence, two such representatives could only be equivalent if the rectangular bands in the same position are. This case is excluded by construction of $\mathcal{C}$.

For a given $\mathcal{D}$-class structure $\{R_\alpha \mid \alpha \in Y\}$ one can assume, without loss of generality, that the underlying sets of the rectangular bands are the ranges from line 9. This leaves only isomorphisms or anti-isomorphisms in $S_n \times C_2$ to be considered for which the bijection in S_n induces a permutation of the rectangular bands, and hence an automorphism or anti-automorphism of Y. A representative of each equivalence class under this induced action is taken in line 13. □

The next step is to add lex-leader constraints (4.3) to each CSP in $\mathcal{L}_{\overline{\mathcal{B}}_n}$ to obtain $\overline{\mathcal{L}}_{\overline{\mathcal{B}}_n}$. According to Lemma 5.1.2(iii) the equivalent solutions of $L_{\{R_\alpha \mid \alpha \in Y\}}$ are orbits under $\mathrm{Stab}_{S_n \times C_2}(\{R_\alpha \mid \alpha \in Y\})$. To calculate the stabiliser in $S_n \times C_2$ of a set of $\mathcal{D}$-classes directly is even more inefficient than the direct stabiliser calculation for diagonals; in particular, because two sets of $\mathcal{D}$-classes $\{D_\beta \mid \beta \in Y\}$ and $\{R_\alpha \mid \alpha \in Y\}$ are identified if there exists an automorphism π of Y such that $D_\beta = R_\alpha$ whenever $\beta^\pi = \alpha$. Recall that in the case of a CSP L_f with fixed entries on the diagonal given by the function f, the stabiliser of the literals given by f was computed. Again, the direct stabiliser calculation can be replaced by calculating the setwise stabiliser of a set of literals using Lemma 5.1.3. All constraints in CSP 5.2.4, but those enforcing associativity, are simple domain restrictions. Consider a CSP $L_{\{R_\alpha \mid \alpha \in Y\}}$. For all table positions (i, j) such that $i, j \in R_\alpha$ for some $\alpha \in Y$, the entries are given by the multiplication in the rectangular bands due to (5.4). The corresponding set of literals is

$$\chi_{\mathrm{I}} = \bigcup_{\alpha \in Y} \{(T_{i,j} = k) \mid i, j, k \in R_\alpha, k = ij\}\,. \tag{5.8}$$

For all other table positions it is only known in which rectangular band the entry has to lie following (5.5). This leads to the set

$$\chi_{\mathrm{II}} = \bigcup_{\substack{\alpha,\beta \in Y \\ \alpha \neq \beta}} \{(T_{i,j} = k) \mid i \in R_\alpha, j \in R_\beta, k \in R_{\alpha \wedge \beta}\} \tag{5.9}$$

of literals. As associativity is invariant under isomorphism and anti-isomorphism, the solutions of $L_{\{R_\alpha|\alpha\in Y\}}$ form orbits under the setwise stabiliser of the literals $\chi_{\mathrm{I}} \cup \chi_{\mathrm{II}}$ in $S_n \times C_2$ by application of Lemma 5.1.3.

Following the comment after Remark 5.2.5 it is clear that $L_{\{R_\alpha|\alpha\in Y\}}$ has exactly one solution if $|Y| = 1$ or $|Y| = n$. These CSPs are therefore excluded from the computation. The subset of $\mathcal{B}_n$ without structures of this kind is denoted by $\mathcal{B}_n^*$. Hence, no instances need to be solved for $n = 1, 2$. The results of solving the instances corresponding to $\mathcal{D}$-class structures in $\mathcal{B}_n^*$ for $3 \leq n \leq 10$ are given in Table 5.3. The numbers for $3 \leq n \leq 8$ agree with the numbers given in [SYT94]. The numbers for $n = 9, 10$ are new results in the enumeration of bands up to equivalence.

Due to the large number of instances in $\overline{\mathcal{L}}_{\overline{\mathcal{B}}_n^*}$, the setup time cannot be neglected in this computation. Nevertheless, the issue that the setup took longer than the solving, as seen in Table 5.3, could most likely be avoided using a more efficient method to create the input files for Minion. No effort was made in this direction, since the restricting factor for the computation is the memory usage. For $n = 10$ the available 16 GB RAM were needed to solve two of the instances. Looking at these two instances in detail one realises that they correspond to the $\mathcal{D}$-class structures consisting of one trivial rectangular band and one left-zero semigroup on $n - 1$ elements. The two possible arrangements are illustrated in Figure 5.1. The bands from these two cases can be classified in general.

Lemma 5.2.8 *For $n \in \mathbb{N}$ with $n \geq 3$, the number of non-equivalent bands of order n whose $\mathcal{D}$-classes are a left-zero semigroup of order $n - 1$ and a trivial semigroup is 1 plus the number of partitions of $n - 1$.*

Proof: Let B be a band from the statement and without loss of generality let its $\mathcal{D}$-classes be $[n - 1]$ and $\{n\}$. There is a unique semilattice on two elements consisting of a zero and an identity. If $\{n\}$ is associated to the zero element in the semilattice, then the product of any element multiplied with n has to lie in $\{n\}$. Hence, the multiplication in B is uniquely determined.

Let now $\{n\}$ be associated to the identity element in the semilattice. Then from $xn = y$ with $y \in [n-1]$ it follows $y = xn = xxn = xy = x$, since $[n-1]$ forms a left-zero semigroup. This means that n is a right identity and that every product equals its left-most factor if this factor does not equal n. Multiplication on the left

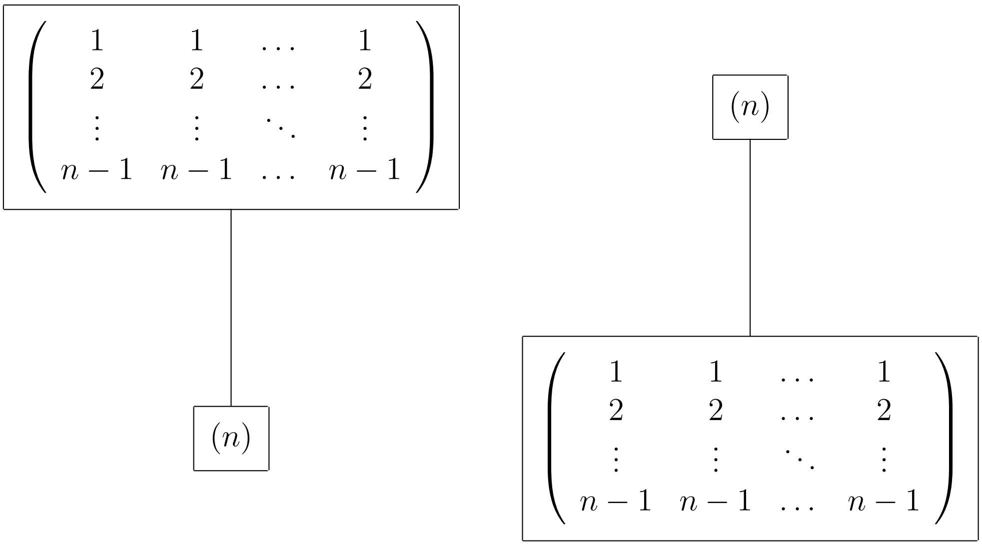

Figure 5.1 The semilattice structures of the two cases in $\overline{\mathcal{L}}_{\overline{\mathcal{B}}_n^*}$ requiring the most memory

with n has to fulfil the idempotent condition $nx = nnx$, but is otherwise arbitrary. Then $n(nx) = (nn)x$ and anyway $n(xy) = nx = (nx)y$, making the multiplication associative.

The band from the first case is not equivalent to any band from the second case, since $n \geq 3$. To count the number of non-equivalent bands from the second case, note that the stabiliser of the $\mathcal{D}$-class structure is $S_{[n-1]}$. As with diagonals one associates a digraph having edges (x, nx) with the row indexed by n. This graph consists of rooted trees of height 1, because of the idempotent condition. Unlabelled graphs of this type are in one-one correspondence with the partitions of $n - 1$. □

The idea used in the proof of the previous lemma is essentially the same as that used by Grillet to describe the first row of a multiplication table [Gri07]. The result can be used to exclude such bands from the search and perform it on a machine with just 1 GB memory for $n = 10$.

The runtime for the family of CSPs $\overline{\mathcal{L}}_{\overline{\mathcal{B}}_n^*}$ suffers so much from the huge number of instances that it is worthwhile to think about a way how to obtain a smaller family still counting all bands. This can be achieved by using only part of the structural information about bands from Lemma 5.2.3. Instead of fixing the whole $\mathcal{D}$-class structure in one CSP just the rectangular band corresponding to the minimal element of the semilattice shall be specified.

CSP 5.2.9 For a rectangular band R on a subset of $[n]$ define a CSP $B_R = (V_n, D_n, C_R)$ based on L_{id_n} by adding the constraints

$$T_{i,j} = ij \quad \text{if} \quad i, j \in R \tag{5.10}$$

$$T_{i,j}, T_{j,i} \in R \quad \text{if} \quad i \in R, j \in [n]. \tag{5.11}$$

The solutions of B_R are exactly the bands on $[n]$ having R as their minimal $\mathcal{D}$-class. Let $\mathcal{R}_n^i$ denote the rectangular bands on any subset of $[n]$ of size i and $\mathcal{R}_n = \cup_{i=1}^n \mathcal{R}_n^i$. Then define the family of CSPs $\mathcal{L}_{\mathcal{R}_n} = \{B_R \mid R \in \mathcal{R}_n\}$. Applying Lemma 5.1.2 yields as before a set $\overline{\mathcal{R}}_n$ of rectangular bands leading to instances with non-equivalent solutions. The set $\overline{\mathcal{R}}_n$ contains each rectangular band of order at most n up to equivalence. The symmetries of an instance B_R are given by the direct product of the stabiliser of R in $S_R \times C_2$ with $S_{[n]\setminus R}$, which again can be computed as stabiliser of the literals.

For some of the CSPs in $\overline{\mathcal{L}}_{\overline{\mathcal{R}}_n}$ the number of solutions is known. Those CSPs with $|R| = n$ have exactly one solution – that is R itself. The number of solutions of the CSP B_R where R is a left zero (or right zero) semigroup on $n-1$ elements equals the number of partitions of $n-1$ according to the proof of Lemma 5.2.8. The results for this case split are shown in Table 5.3. One can see that the smaller number of instances yields a big improvement for orders up to 9. The larger number of symmetries prevents any result for $n = 10$. With regard to the enumeration of semigroups of order 9, of the presented approaches this one is the most efficient to count the bands with 9 elements up to equivalence.

5.3 52 989 400 714 478

In the last section the number of solutions of the single instance in $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_9}^{-3}$ which could not be solved directly was determined using another case split. Counting all solutions of instances in $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_9} \setminus \overline{L}_{\mathrm{id}_9}$ with Minion, version 0.9, on a machine with 2.66 GHz Intel X-5430 processor and 8 GB RAM took around 87 hours and output 23 161 033 393 as result. Together with the number of bands of order 9 from Table 5.3, the number of 3-nilpotent semigroups of order 9, computed using the formula from Theorem 2.3.7, and the single 2-nilpotent semigroup, this yields the

Table 5.4 Numbers of non-equivalent semigroups on $[n]$ by idempotent

n / Idpt.	1	2	3	4	5	6	7	8	9
1	1	2	5	19	132	3 107	623 615	1 834 861 133	52 976 551 026 562
2		2	7	37	216	1 780	32 652	4 665 709	12 710 266 442
3			6	44	351	3 093	33 445	600 027	68 769 167
4				26	326	4 157	53 145	754 315	14 050 493
5					135	2 961	56 020	1 007 475	18 660 074
6						875	30 395	822 176	20 044 250
7							6 749	348 692	12 889 961
8								60 601	4 389 418
9									618 111
$\sum$	1	4	18	126	1 160	15 973	836 021	1 843 120 128	52 989 400 714 478

number of semigroups of order 9 up to equivalence. There are 52 989 400 714 478 non-equivalent semigroups with 9 elements. The numbers of semigroups with at most 9 elements sorted by their number of idempotents are listed in Table 5.4.

Information on the classification of semigroups of order 9 in the form of [SYT94, Table 4.2] is summarised in Table 5.5. The constraints added to obtain commutative semigroups are obviously $T_{i,j} = T_{j,i}$ for all $i, j \in [n]$. A semigroup S is *regular* if for all $s \in S$ there exists a $t \in S$ such that $sts = s$. If in addition all idempotents in S commute then S is *inverse*. This leads to the constraints

$$\forall i \in [n] \exists j \in [n] : T_{T_{i,j},i} = i \tag{5.12}$$

to get the numbers of regular semigroups and in addition

$$T_{i,j} = T_{j,i} \text{ for } i, j \in \{k \in [n] \mid f(k) = k\} \tag{5.13}$$

to get the numbers of inverse semigroups. How self-dual semigroups are determined has been discussed in the first paragraph of Section 4.2.2.

For order 9 the percentage of semigroups that are 3-nilpotent reaches 99.96 % and is thereby getting even closer to 100 % than for order 8. These numbers support the conjecture that the ratio converges to 1 while n tends to infinity.

Other than for orders up to 8 the multiplication tables were counted but not stored. In principle, the code provided on the attached DVD can be used to

Table 5.5 Properties of semigroups of order 9

Idpt.	self-dual	commutative	regular	inverse	comm.-inv.
1	613 365 656	9 940 825	2	2	2
2	8 265 721	664 080	23	23	16
3	739 317	249 330	148	129	111
4	410 158	222 637	830	567	504
5	328 937	201 060	4 136	1 750	1 555
6	223 226	148 647	17 535	3 870	3 460
7	113 160	82 481	66 822	6 582	6 137
8	38 979	30 789	217 437	7 505	7 505
9	7 510	5 994	618 111	5 994	5 994
$\sum$	623 492 664	11 545 843	925 044	26 422	25 284

obtain multiplication tables of semigroups of order 9. Note that the output of Minion for $2.3 * 10^{10}$ tables of dimension 9 will occupy more than 3.5 TB of disk space. Moreover, the 3-nilpotent semigroups of order 9 were not even counted using Minion. Extrapolation from tests on the number of solutions, which Minion counts per second for such instances – roughly 1 million, suggests that the computation time on a 2.66 GHz Intel X-5430 processor would be between 1 and 2 years.

In the following further refinements are presented which aim in particular to reduce the memory required for the computation. This is achieved by further case splits, leading to a smaller number of lex-leader constraints (4.3) to be posted in each CSP, and fixing more entries of the multiplication table. This allows one to determine the number of semigroups with 9 elements on a 32-bit machine with as little as 1 GB of memory. Furthermore, it introduces case splits that are useful for the enumeration of subclasses of semigroups for orders higher than 9.

5.3.1 Constant function

In the following let c be the constant function $c : [n] \to [n], i \mapsto n$. With $2(n-1)!$ lex-leader constraints $\overline{L}_c$ is the CSP in $\overline{\mathcal{L}}_{\mathcal{F}_n} \setminus \{\overline{L}_{\mathrm{id}_n}\}$ leading to the instance with the biggest memory usage.[2] The stabiliser of c in $S_n \times C_2$ is $S_{n-1} \times C_2$. Under this group the off-diagonal positions in the last row and column of the multiplication

[2]In fact, the representative from the orbit of constant functions returned by the implementation of Algorithm 1 into GAP is not c, but the function with image $\{1\}$. Using c here simplifies some of the notation.

table form a separate orbit. Since in L_c the implied constraint (5.2) becomes $T_{n,i} = T_{i,n}$ for all $i \in [n]$, the last row and column are equal in all solutions of L_c.

CSP 5.3.1 Let $L_f \in \mathcal{L}_{\mathcal{F}_n}$ and let $p : [n-1] \to [n-1]$ be a partial function. Then define a CSP L_f^p based on L_f from CSP 5.1.1 by adding for all $i \in [n]$ the constraint

$$T_{n,i} = \begin{cases} p(i) & \text{if } p \text{ is defined on } i \\ n & \text{otherwise.} \end{cases}$$

Let $\mathcal{P}_{n-1}$ denote all partial functions on $[n-1]$. Employing the correspondence between partial functions and digraphs, a set of non-equivalent partial functions $\overline{\mathcal{P}}_{n-1}$ under the action of S_{n-1} was determined in Section 3.1.

Only CSPs L_c^p with $p \in \mathcal{P}_{n-1}$ where p does respect the idempotent condition $T_{n,i} = T_{n,T_{n,i}}$ will possibly have solutions. Denote the set of those partial function by $\mathcal{P}^*_{n-1}$. Which digraphs correspond to partial functions satisfying the idempotent condition? Let $p \in \mathcal{P}^*_{n-1}$ be a partial function. If $i \in [n-1]$ is in the image of p, then $p(i) = i$. It follows that each cycle in the corresponding digraph is a loop and the tree rooted at the loop vertex has height at most 1. Further, each $i \in [n-1]$ on which p is undefined has to label an isolated vertex without any edges. Hence, such a graph is uniquely determined by an integer $0 \leq k \leq n-1$ specifying the number of isolated vertices and a partition of $n-1-k$, each summand specifying the number of vertices in one of the remaining components. These facts were used to construct the partial functions in $\overline{\mathcal{P}}^*_{n-1}$.

The stabiliser of the partial function $p \in \mathcal{P}^*_{n-1}$ in $S_{n-1} \times C_2$ are the symmetries for which lex-leader constraints (4.3) are added to L_c^p to obtain $\overline{L}_c^p$. Hence, most CSPs in the family $\left\{\overline{L}_c^p \mid p \in \overline{\mathcal{P}}^*_{n-1}\right\}$ replacing $\overline{L}_c$ have far fewer constraints. The only exception is the partial function $p_\emptyset$ which is nowhere defined. The CSP $\overline{L}_c^{p_\emptyset}$ is special as all its solutions lead to nilpotent semigroups. This follows immediately from Lemma 2.1.1, since n is a zero element and the square of each element equals n.

No CSP in $\left\{\overline{L}_c^p \mid p \in \overline{\mathcal{P}}^*_{n-1} \setminus \{p_\emptyset\}\right\}$ has more than $2(n-2)!$ lex-leader constraints. For $n = 9$ the solution numbers, 16 512 454 in total, for the corresponding instances could be obtained on a machine with 1 GB RAM. It remains to find the solutions of $\overline{L}_c^{p_\emptyset}$.

The CSP $\overline{\mathbf{L}}_{\mathbf{c}}^{\mathbf{p}\emptyset}$

The fact that all solutions of $\overline{L}_c^{p\emptyset}$ are nilpotent semigroups can be employed for another case split.

Definition 5.3.2 Let S be a nilpotent semigroup with zero element z, and let $a \in S \setminus \{z\}$. If $as = sa = z$ for all $s \in S$, then a is an *annihilator*.

Every r-nilpotent semigroup S contains at least one annihilator, that is any element in the set $S^{r-1} \setminus S^r$, which is non-empty according to Lemma 2.1.2(i).

CSP 5.3.3 For a non-empty subset $A \subseteq [n-1]$ define a CSP L_c^A based on $L_c^{p\emptyset}$ by adding, for all $i \in A$ and for all $j \in [n]$, the constraints $T_{i,j} = n$ and $T_{j,i} = n$, and for all $i \in [n-1] \setminus A$ the constraint

$$\exists j \in [n] : T_{i,j} \neq n \text{ or } T_{j,i} \neq n.$$

The constraints in L_c^A ensure that in any solution precisely the elements in A are annihilators. The equivalent solutions of L_c^A are the orbits under the action of the group $S_A \times S_{[n-1]\setminus A} \times C_2$. Since the stabiliser of c in S_n is S_{n-1}, CSPs with the same number of annihilators have equivalent solution tables. This yields a CSP family in the same fashion as for the case split on the number of idempotents used to obtain the results in Table 5.2. The CSP $\overline{L}_c^{p\emptyset}$ has the same solutions as the family of CSPs $\{\overline{L}_c^A \mid A = [m], 1 \leq m \leq n-1\}$.

Indeed, the presented approach works for all functions f leading to nilpotent semigroups; a condition that can be checked using the criterion from Lemma 3.2.5. Obtaining the family of independent CSPs will be slightly different though. Since the stabiliser for $f \neq c$ will not be S_{n-1}, CSPs with the same number of annihilators do not necessarily have equivalent solutions. Furthermore, the number of annihilators is restricted by the size of the preimage of the fixed point of f.

Details of the generalisation of CSP 5.3.3 to non-constant functions are not given, because the case split for $f \neq c$ does not reduce the maximal amount of memory needed to solve the instances in $\overline{\mathcal{L}}_{\mathcal{F}_n}$. With $4(n-2)!$ lex-leader constraints the next bigger instance arises from the CSP $\overline{L}_f$, where f has $n-2$ fixed points and the remaining two points are mapped to each other. The digraph corresponding to f consists of $n-2$ loops and one cycle of length 2. According to Lemma 3.2.7 elements labelling the vertices in a cycle lie in a common subgroup. Without loss of generality, one can choose any fixed point to be the identity in the group containing the elements labelling the 2-cycle. That there is a unique group of order 3 with a given diagonal determines 6 further entries, and reduces the number of symmetries to $(4(n-3)!)$.

Further refinements are not explained as they had no considerable effects at this stage. Some possible ideas are discussed in Section 5.6.

5.4 The Monoids of Order at most 10

There are 858 977 non-equivalent monoids of order 8, 1 844 075 697 of order 9, and 52 991 253 973 742 of order 10. These numbers have first been published by the author and Kelsey in [DK08], respectively [DK09], and this section reports on their findings.

The counting of bands in Section 5.2 showed that the enumeration using constraint satisfaction can successfully be applied to subclasses of semigroups. The search benefited hugely from the structural knowledge about bands incorporated in the CSP model.

Monoids are another important subclass of semigroups, and there exist results on the structure of finite monoids, which can be exploited for the search. Before the work on this thesis started, monoids had only been enumerated up to order 7 – though obviously the counting of semigroups of order 8 in [SYT94] included monoids.

Adapting the CSP family $\overline{\mathcal{L}}_{\mathcal{F}_n}$ from Section 5.1, the numbers of monoids up to order 9 were determined in [DK08]. This approach is presented in the forthcoming subsection.

Monoids of order 10 have first been enumerated in [DK09] using a refined CSP model which facilitates structural information about finite monoids. In Section 5.4.2 the structure of finite monoids is discussed, and the refined CSP model is explained.

The presentation of the results from [DK08] and [DK09] is adjusted for this thesis. Moreover the computations were repeated using the latest version of Minion leading to considerable speed-ups.

5.4.1 Basic CSP and diagonal case split

The basic CSP model L_n to find all semigroups on $[n]$ from CSP 4.1.3 is modified to return all monoids on $[n]$ by adding the constraint

$$\exists i \in [n] \forall j \in [n] : T_{i,j} = T_{j,i} = j. \tag{5.14}$$

When aiming to find monoids up to equivalence one can assume without loss of generality that n is the identity element.[3] Then (5.14) becomes

$$\forall j \in [n] : T_{n,j} = T_{j,n} = j. \tag{5.15}$$

Denote the new CSP with M_n. Formally, M_n was obtained by application of Lemma 5.1.2 to the family of n CSPs fixing 1 through n to be the identity element. That the identity in a monoid (even in a magma) is unique ensures these instances are disjoint. The induced action on the identity is the natural action of S_n, and the stabiliser of n in S_n is S_{n-1}. Hence the solutions of M_n form orbits under $S_{n-1} \times C_2$ according to Lemma 5.1.2(iii).

Next, the idea from Section 5.1 is adapted to split M_n into a family of CSPs depending on the diagonal of the multiplication table. The set of diagonals used to get instances with non-equivalent solutions will differ from the set used for the family $\mathcal{L}_{\overline{\mathcal{F}}_n}$, because the group defining the equivalence of diagonals is no longer S_n but S_{n-1}. It shall be used that functions from $[n]$ to $[n]$ with n as fixed point are in one-one correspondence with partial functions from $[n-1]$ to $[n-1]$. If p is such a partial function, the corresponding function on $[n]$ maps every $i \in [n]$ for which p is undefined to n.

CSP 5.4.1 Let $p : [n-1] \to [n-1]$ be a partial function. Then define a CSP M_p by adding constraints (5.1) to M_n for the function f corresponding to p. That is for all $i \in [n]$

$$T_{i,i} = \begin{cases} p(i) & \text{if } p \text{ is defined on } i \\ n & \text{otherwise.} \end{cases}$$

Recall that $\mathcal{P}_{n-1}$ denotes the set of all partial functions on $[n-1]$. How to get a set $\overline{\mathcal{P}}_{n-1}$ of partial functions on $[n-1]$ up to S_{n-1}-equivalence was discussed in

[3] As for the constant function in Section 5.3.1 this choice is made for simplicity of notation. In the computation 1 is the identity element.

Section 3.1 (It should make sense now, why the set of diagonals obtained in this way is useful.). One defines a family of CSPs $\mathcal{M}_{\overline{\mathcal{P}}_{n-1}} = \left\{M_p \mid p \in \overline{\mathcal{P}}_{n-1}\right\}$ in which distinct CSPs have non-equivalent monoids as solutions and each type of monoid appears as solution of one CSP. The solutions of one of the CSPs M_p form orbits under the stabiliser of the partial function p in $S_{n-1} \times C_2$. Applying Lemma 5.1.3 again, the stabiliser can be computed efficiently as the setwise stabiliser of the literals $\{(T_{i,i} = p(i)) \mid i \in [n], p(i) \text{ is defined}\}$.

A simple observation helps making the family $\mathcal{M}_{\overline{\mathcal{P}}_{n-1}}$ far more efficient. One constructs a set of non-equivalent monoids of order n by taking the semigroups on $[n-1]$ and adding n as new identity element. A monoid M with identity e is equivalent to one constructed in this way if and only if $M \setminus \{e\}$ forms a semigroup. Since non-equivalent semigroups are known from Section 5.1, it is not necessary to search for these monoids. Excluding them from the search translates to the condition that the identity appears at least twice (once from $ee = e$) in the multiplication table. Denote the CSPs excluding such monoids by M_p^e, and let $\mathcal{M}^e_{\overline{\mathcal{P}}_{n-1}}$ denote the corresponding family of CSPs $\left\{M_p^e \mid p \in \overline{\mathcal{P}}_{n-1}\right\}$.

Solving either the basic model $\overline{M}_n$ or the family of CSPs $\overline{\mathcal{M}}_{\overline{\mathcal{P}}_{n-1}}$ yields the number of monoids for orders $n, 1 \leq n \leq 9$ (see Table 5.6). Solving $\overline{M}^e_n$, respectively $\overline{\mathcal{M}}^e_{\overline{\mathcal{P}}_{n-1}}$, is much faster and results in the number of monoids on $[n]$ minus the number of semigroups on $[n-1]$. The numbers for monoids with 8 and 9 elements were first published in [DK08] using essentially the same family of CSPs. Refinements excluding certain partial functions p from search, for which M_p^e has no solutions, are explained in the cited publication. No further details are given here, since these limited considerations will become obsolete, after learning more about the structure of finite monoids in the forthcoming subsection.

In [KR85, Theorem 2] Koubek and Rödl proved that for unlabelled monoids the number of those monoids, that are constructed by adding a new identity to a semigroup, yields an asymptotic bound for the number of all monoids.[4] The computational results suggest that the same holds for monoids up to equivalence.

[4] Note that these results are based on [KRS76], in which details in the proof of the main result are omitted.

Table 5.6 Enumeration of non-equivalent monoids on $[n]$

n	1	2	3	4	5	6	7	8	9
$\overline{M}_n$, solutions	1	2	6	27	156	1 373	17 730	858 977	1 844 075 697
–, setup time	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	4 s	51 s	588 s
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	2 s	133 s	934 261 s
$\overline{M}^e_n$, solutions	0	1	2	9	30	213	1 757	22 956	955 569
–, setup time	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	4 s	52 s	624 s
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	1 s	11 s	737 s
$\overline{\mathcal{M}}_{\overline{\mathcal{P}}_{n-1}}$, solutions	1	2	6	27	156	1 373	17 730	858 977	1 844 075 697
–, instances	1	2	6	16	45	121	338	929	2 598
–, setup time	ϵ	ϵ	ϵ	ϵ	1 s	9 s	53 s	362 s	3 076 s
–, solve time	ϵ	ϵ	ϵ	ϵ	1 s	3 s	12 s	78 s	19 986 s
$\overline{\mathcal{M}}^e_{\overline{\mathcal{P}}_{n-1}}$, solutions	0	1	2	9	30	213	1 757	22 956	955 569
–, instances	1	2	6	16	45	121	338	929	2 598
–, setup time	ϵ	ϵ	ϵ	ϵ	1 s	9 s	55 s	459 s	3 220 s
–, solve time	ϵ	ϵ	ϵ	ϵ	1 s	2 s	10 s	47 s	289 s

The times are rounded to seconds. They were obtained using the 64-bit executable of Minion version 0.9 on a machine with 2.66 GHz Intel X-5430 processor and 16 GB RAM. The setup was done with GAP. The symbol ϵ stands for a time less than 0.5 s.

5.4.2 Structure of finite monoids

Similar to the enumeration of bands, structural information about monoids shall be used to improve the CSP model.

Definition 5.4.2 Let M be a monoid and let e be its identity. The elements in H_e, the $\mathcal{H}$-class of e, are called *units*.

Since the units of a monoid are the $\mathcal{H}$-class of an idempotent, they form a group (see [How95, Theorem 2.2.5]), the *unit group*. It is easy to check from the multiplication table whether two elements lie in the same $\mathcal{H}$-class. Each of the two has to appear in the column and in the row of the other. Thus, an element in a monoid is a unit if and only if the identity appears in its row and column. This means any unit has a left and a right inverse, and hence, in fact, a unique inverse. Thus units are precisely the invertible elements in a monoid.

Information about the multiplication of elements outside the unit group is known as well. The following lemma provides a structural description for finite monoids.

Lemma 5.4.3 *Let M be a finite monoid and let e be its identity. Then the following hold:*

(i) *the set $M \setminus H_e$ forms the unique maximal ideal of M;*

(ii) *multiplication from the left of elements in the unit group H_e on $M \setminus H_e$,*

$$\lambda_M : H_e \times M \setminus H_e \to M \setminus H_e, (u, m) \mapsto um, \tag{5.16}$$

defines a left action of H_e on the maximal ideal;

(iii) *multiplication from the right of elements in the unit group H_e on $M \setminus H_e$,*

$$\rho_M : M \setminus H_e \times H_e \to M \setminus H_e, (m, u) \mapsto mu \tag{5.17}$$

defines a right action of H_e on the maximal ideal.

Proof: (i): It is shown in [Gri95, Chapter V, Proposition 1.3] that $M \setminus H_e$ forms an ideal. Let I be an ideal of M containing a unit u. Then for all $m \in M$ the product

$u^{-1}um$ lies in I. Consequently $I = M$. Hence $M \setminus H_e$ is the unique maximal ideal of M.

(ii): Since $M \setminus H_e$ is an ideal, the map λ_M is well-defined. The identity e of M acts trivially by definition. Furthermore for $g, h \in H_e$ the equality

$$(g, (h, m)) = g(hm) = (gh)m = (gh, m)$$

holds for all $m \in M \setminus H_e$ because of associativity of M. Hence λ_M is a left action.

(iii): The proof is analogous to that for the second part. □

Following Lemma 5.4.3 every finite monoid M decomposes uniquely into a group, its unit group, and a semigroup, its maximal ideal. Note that not every left and right action of a group G on a semigroup S yield a monoid on $G \cup S$. On the other hand, if a monoid is defined, then it has indeed G as its unit group, since G is then the $\mathcal{H}$-class of the identity.

Remark 5.4.4 Given a group $(G, *_G)$ and a semigroup $(S, *_S)$ on disjoint sets G and S, define a multiplication $*$ on $G \cup S$ using the trivial left and right action of G on S. That is

$$x * y = \begin{cases} x *_G y & \text{if } x, y \in G \\ x & \text{if } x \in S, y \in G \\ y & \text{if } x \in G, y \in S \\ x *_S y & \text{if } x, y \in S. \end{cases}$$

Since in every mixed product the factors from G are ignored, $(G \cup S, *)$ is a monoid.

A monoid where the left and right action of the unit group on the maximal ideal are trivial will be referred to as *trivial action monoid.*

The next result on equivalent trivial action monoids looks on the first glance like an obvious result about isomorphic structures. One must not forget that equivalence adds a subtle point to this type of considerations. That the following lemma holds, is therefore not clear a priori.

Lemma 5.4.5 *Let $(M, *_M)$ with identity e and $(N, *_N)$ with identity f be two trivial action monoids. Then M and N are equivalent if and only if their unit groups H_e and H_f are isomorphic groups and the maximal ideals $M \setminus H_e$ and $N \setminus H_f$ are equivalent semigroups.*

Proof: $(\Rightarrow)$: Let first $\sigma : M \to N$ be an isomorphism or anti-isomorphism. Then σ maps e to f and hence H_e to H_f. Therefore the restrictions of σ on H_e respectively $M \setminus H_e$ are isomorphisms or anti-isomorphisms. That anti-isomorphic groups are in fact isomorphic completes the proof for this direction.

$(\Leftarrow)$: Let $\sigma_{H_e} : H_e \to H_f$ and $\sigma_{M\setminus H_e} : M \setminus H_e \to N \setminus H_f$ be isomorphisms. Then $\sigma : M \to M$ defined by

$$\sigma(m) = \begin{cases} \sigma_{H_e}(m) & \text{if } m \in H_e \\ \sigma_{M\setminus H_e}(m) & \text{otherwise} \end{cases}$$

defines an isomorphism. If on the other hand $\sigma_{M\setminus H_e} : M \setminus H_e \to N \setminus H_f$ is an anti-isomorphism, then an anti-isomorphism $\sigma : M \to M$ is given by

$$\sigma(m) = \begin{cases} (\sigma_{H_e}(m))^{-1} & \text{if } m \in H_e \\ \sigma_{M\setminus H_e}(m) & \text{otherwise,} \end{cases}$$

where the inverse is taken in the group H_f. □

Due to the previous result the number of non-equivalent trivial action monoids can be expressed in terms of the the number of non-isomorphic groups and non-equivalent semigroups.

Corollary 5.4.6 *Let $n \in \mathbb{N}$. If g_k denotes the number of non-isomorphic groups, and h_k the number of non-equivalent semigroups on $[k], k \in \mathbb{N}$, then*

$$g_n + \sum_{i=1}^{n-1} g_i h_{n-i}$$

equals the number of non-equivalent trivial action monoids on $[n]$.

For some sizes of group and semigroup the only possible actions are trivial.

Lemma 5.4.7 *Let M be a monoid with n elements and $1, n-1$ or n units. Then M is a trivial action monoid.*

Proof: If M contains only one unit, then it is the identity element, which always acts trivially. If M contains $n-1$ units, then there is only one element that is not a unit, and every action on a set with one element is trivial. There is nothing to show if M is a group. □

The structure theorem for monoids could be used to define a family of CSPs, each specifying the unit group and the maximal ideal. While every instance would be easy to solve, their number would be too big to make this approach practicable. The aim of using more of the structure of monoids is to enumerate at least the monoids with 10 elements. If then the unit group was of order 2, one would have to solve one CSP for each non-equivalent semigroup of order 8, of which there are 1 843 120 128.

To avoid the problem of having to solve too many instances, a family of CSPs using only part of the structural information about monoids from Lemma 5.4.3 is introduced. As there are far fewer groups than semigroups the idea is to fix the unit group of the monoids to be searched for, but not the maximal ideal.

CSP 5.4.8 For a group $(G, *_G)$ with identity e and $G \subseteq [n]$ define a CSP $M_G = (V_n, D_n, C_G)$ based on L_n from CSP 4.1.3. Add the following constraints to C_n to obtain C_G:

$$T_{i,j} = i *_G j \quad \text{for all} \quad i, j \in G \tag{5.18}$$

$$T_{e,j} = T_{j,e} = j \quad \text{for all} \quad j \in [n] \setminus G \tag{5.19}$$

$$T_{i,j}, T_{j,i} \in [n] \setminus G \quad \text{for all} \quad i \in [n], j \in [n] \setminus G. \tag{5.20}$$

Constraint (5.18) fixes entries in the multiplication table indexed by elements in G according to the multiplication in the group. Then constraint (5.19) ensures that e is an identity element in every solution, which is therefore a monoid. Elements outside G are not units due to constraint (5.20). Hence the solutions of M_G are the monoids on $[n]$ having G as unit group. That the elements in G are units leads to the implied constraints

$$\forall i \in G \forall j, k \in [n] \setminus G \text{ with } j \neq k : T_{i,j} \neq T_{i,k} \;\&\; T_{j,i} \neq T_{k,i},$$

which require the entries in a row and column of a unit to be all different.

The stabiliser of G in $S_n \times C_2$ is a subgroup of $S_G \times S_{[n]\setminus G} \times C_2$ and can by application of Lemma 5.1.3 be computed using the literals of M_G. Adding lex-leader constraints (4.3) for all elements in the stabiliser yields $\overline{M}_G$. If $\mathcal{G}_{\leq n}$ denotes all groups on $[k]$ for $1 \leq k \leq n$, then $\overline{\mathcal{G}}_{\leq n}$ are the groups of order at most n up to isomorphism. The family of CSPs $\overline{\mathcal{M}}_{\overline{\mathcal{G}}_{\leq n}} = \left\{\overline{M}_G \mid G \in \overline{\mathcal{G}}_{\leq n}\right\}$ has as solutions the monoids on $[n]$ up to equivalence.

The results from Table 5.6 show that most monoids are trivial action monoids. To avoid searching for those, add the constraint

$$\exists i \in G, j \in [n] \setminus G : (T_{i,j}, T_{j,i}) \neq (j, j) \tag{5.21}$$

to M_G to obtain M_G^{nta} (for *n*on-*t*rivial *a*ction) and respectively the family $\overline{\mathcal{M}}^{nta}_{\overline{\mathcal{G}}_{\leq n}}$.

From the experience of the enumeration of semigroups it seems desirable to have some further case split for the maximal ideal. Adopting the idea in Section 5.1, the part of the diagonal, which does not belong to the unit group shall be fixed in addition.

CSP 5.4.9 Based on M_G from CSP 5.4.8 define a CSP $M_{G,f} = (V_n, D_n, C_{G,f})$ for a function $f : [n] \setminus G \to [n] \setminus G$ by adding the constraints $T_{i,i} = f(i)$ for all $i \in [n] \setminus G$ to C_G to obtain $C_{G,f}$.

The group $S_n \times C_2$ acting on multiplication tables induces a componentwise action on the pairs (G, f). Two pairs are equivalent if the groups are isomorphic and the functions are equivalent. The stabiliser of (G, f) in $S_n \times C_2$ is the direct product of the stabiliser of $S_G \times C_2$ with the stabiliser of f in $S_{[n]\setminus G}$.

The functions corresponding to diagonals of semigroups are known up to equivalence from Section 3.2. Instead of the set of all functions on $[n]$ up to equivalence, $\overline{\mathcal{F}}_n$, this smaller set, denoted by $\overline{\mathcal{F}}^*_n$, can be used. Then $\overline{\mathcal{M}}_{\overline{\mathcal{G}}_{\leq n}, \overline{\mathcal{F}}^*_{\leq n}} = \left\{\overline{M}_{G,f} \mid G \in \overline{\mathcal{G}}_{\leq n}, f \in \overline{\mathcal{F}}^*_{n-|G|}\right\}$ has again all monoids on $[n]$ up to equivalence as solutions. Adding constraint (5.21) excludes trivial action monoids as solutions. This makes it in particular unnecessary to consider unit groups of size 1, $n-1$ or n (see Lemma 5.4.7). The results of solving the corresponding instances for $1 \leq n \leq 10$ are given in Table 5.7, showing a considerable speed-up compared with the approach fixing only the diagonal of the monoid. The number of monoids on 10 elements has first been published by the author and Kelsey in [DK09].

Note that 16 GB memory were needed to run the Minion computation for the group on 2 elements and the identity function on 8 points. This made it impossible to run the computation in this way for $n = 11$ with the available resources. On the

Table 5.7 Enumeration of non-equivalent monoids up to order 10

n	1	2	3	4	5	6	7	8	9	10
$\overline{\mathcal{M}}^{e}_{\overline{\mathcal{G}}_{\leq n}}$, solutions	0	1	2	9	30	213	1 757	22 956	955 569	1 853 259 264
–, instances	0	2	3	5	6	8	9	14	16	18
–, setup time	–	ϵ	ϵ	ϵ	ϵ	ϵ	1 s	7 s	62 s	697 s
–, solve time	–	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	5 s	233 s	877 918 s
$\overline{\mathcal{M}}^{nta}_{\overline{\mathcal{G}}_{\leq n}}$, solutions	0	0	0	2	5	58	428	5 539	101 082	9 269 715
–, instances	0	0	0	1	2	4	5	7	8	13
–, setup time	–	–	–	ϵ	ϵ	ϵ	1 s	6 s	61 s	701 s
–, solve time	–	–	–	ϵ	ϵ	ϵ	ϵ	2 s	48 s	5 764 s
$\overline{\mathcal{M}}_{\overline{\mathcal{G}}_{\leq n},\overline{\mathcal{F}}^{*}_{\leq n}}$, solutions	1	2	6	27	156	1 373	17 730	858 977	1 844 075 697	*52 991 253 973 742*
–, instances	1	2	4	10	21	49	112	261	599	1 384
–, setup time	ϵ	ϵ	ϵ	ϵ	ϵ	3 s	18 s	117 s	964 s	$\diamond$
–, solve time	ϵ	ϵ	ϵ	ϵ	ϵ	1 s	7 s	55 s	20 035 s	$\diamond$
$\overline{\mathcal{M}}^{e}_{\overline{\mathcal{G}}_{\leq n},\overline{\mathcal{F}}^{*}_{\leq n}}$, solutions	0	1	2	9	30	213	1 757	22 956	955 569	1 853 259 264
–, instances	0	1	2	5	10	23	52	123	280	644
–, setup time	–	ϵ	ϵ	ϵ	ϵ	1 s	7 s	33 s	195 s	1 317 s
–, solve time	–	ϵ	ϵ	ϵ	ϵ	ϵ	2 s	11 s	96 s	27 526 s
$\overline{\mathcal{M}}^{nta}_{\overline{\mathcal{G}}_{\leq n},\overline{\mathcal{F}}^{*}_{\leq n}}$, solutions	0	0	0	2	5	58	428	5 539	101 082	9 269 715
–, instances	0	0	0	2	7	20	49	117	273	640
–, setup time	–	–	–	ϵ	ϵ	1 s	6 s	31 s	186 s	1 302 s
–, solve time	–	–	–	ϵ	ϵ	ϵ	2 s	8 s	60 s	963 s

The times are rounded to seconds. They were obtained using the 64-bit executable of Minion version 0.9 on a machine with 2.66 GHz Intel X-5430 processor and 16 GB RAM. The setup was done with GAP. The symbol ϵ stands for a time less than 0.5 s. The symbol $\diamond$ indicates insufficient memory.

other hand, the identity function makes the maximal ideal a band. As there are relatively few bands (compared with the number of all semigroups) the approach discarded before, specifying the unit group and the maximal ideal, becomes more feasible.

It seems plausible that the approach just described would yield the number of monoids on 11 elements that are not trivial action monoids, using the available resources. As the number of semigroups of order 10 is not known at the moment this would not yet result in knowledge about all monoids on 11 elements. Therefore priority was given to other computations such as the identification of automorphism groups of semigroups of order 9 in the next section.

5.5 Automorphism Groups

So far the emphasis was on the enumeration of semigroups or subclasses of semigroups using constraint satisfaction. At some points along the way it was mentioned that additional properties were determined. An enumeration of commutative, regular and inverse semigroups by the number of idempotents is given in Table 5.5. Moreover, the self-dual semigroups were identified using a modified CSP described in Section 4.2.2.

Recall that a semigroup is self-dual if it allows an anti-automorphism, that is an anti-isomorphism mapping the semigroup to itself. In the CSP to find all self-dual semigroups an additional constraint enforced equality for at least one of the lex-leader constraints (4.3) that belong to an anti-isomorphism. The same principal can be applied to determine or count semigroups with prescribed automorphisms. A bijection is an automorphism if equality holds in the corresponding lex-leader constraint (4.3). Several cases are distinguished depending on the orders of automorphisms.

$\mathbf{Aut(S)} \cong \mathbf{C_2}$ Require strict inequality in all lex-leader constraints that do not correspond to a permutation of order 2. Require that equality holds for exactly one of the remaining lex-leader constraints.

$\mathbf{Aut(S)} \cong \mathbf{C_2^2}$ Require strict inequality in all lex-leader constraints that do not correspond to a permutation of order 2. Require that equality holds for exactly three of the remaining lex-leader constraints.

$\mathbf{Aut(S) \cong C_2^k, k \geq 3}$ Require strict inequality in all lex-leader constraints that do not correspond to a permutation of order 2. Require that equality holds for at least four of the remaining lex-leader constraints.

$\mathbf{|Aut(S)| = 2^k, Aut(S) \ncong C_2^k}$ Require strict inequality in all lex-leader constraints that do not correspond to a permutation of order $2^m, m \in \mathbb{N}$. Require that equality holds for at least one lex-leader constraint corresponding to a permutation of order 4.

$\mathbf{|Aut(S)| \neq 2^k}$ Require that equality holds for at least one lex-leader constraint corresponding to a permutation of odd prime order.

These five distinct cases cover any non-trivial automorphism group. In the first two cases, where the isomorphism type of the automorphism group is uniquely determined, the number of solutions is returned by Minion. In the remaining three cases a list of automorphisms for each solution table is output and read into GAP. In the third case the length of each list determines the automorphism group of the corresponding solution, while in the other two cases the identification function in the SmallGroups library [BEO02] is used to find the isomorphism types. In Table 5.8 the numbers of semigroups with trivial automorphism group is compared with the number of semigroups with C_2 as automorphism group and with the remaining semigroups. The detailed list of all automorphism groups was put into Appendix A.2, because of its volume. The Minion computations to obtain the results were the most expensive calculations undertaken in the course of this thesis and took nearly two months on a machine with 2.66 GHz Intel X-5430 processor. As a consequence the results for semigroups of order 9 have not been verified in a second run.

Table 5.8 Automorphism groups of non-equivalent semigroups S on $[n]$

n / Aut(S)	1	2	3	4	5	6	7	8	9
trivial	1	3	12	78	746	10 965	746 277	1 834 638 770	52 961 873 376 696
C_2	0	1	5	39	342	4 121	76 704	8 176 697	27 478 363 462
other	0	0	1	9	72	887	13 040	304 661	48 974 320

Araújo, von Bünau, Mitchell and Neunhöffer implemented an algorithm for the computation of automorphism groups of semigroups into GAP, which is described in [ABMN09]. They used the implementation to determine automorphism groups of semigroups of order at most 7. An obvious omission in [ABMN09] of $C_2 \times C_2$ as automorphism group for semigroups of order 5 is corrected in Appendix A.2. Apart from this exception the results coincide with the data from the reference.

5.6 Outlook

In this final section of the thesis future applications of the methods developed in the previous and in this chapter are outlined; possible extensions and the impact on research are discussed.

5.6.1 Semigroups of order 10

After presenting the number of semigroups of order 9, the most natural question is whether the semigroups with 10 elements can be enumerated using the same method. This is in first place a question about computational resources – both time and space – that are required. Using the approach based on Lemma 5.1.2 of splitting any instance which would need too much memory into a family of smaller instances avoids problems with space. Certainly, this technique has its limitations as well, but first tests showed that it does work for semigroups of order 10. It was possible to set up all instances and to determine for each of them whether it has any solution on a machine with 16 GB RAM.

The other limited resource for computations is time. Extrapolation from the fastest solving of instances for order 9 indicates that it would take at the same speed at least 10 years to enumerate semigroups of order 10. As it becomes more and more common to have multiple processors in one machine or even to use clusters for all kinds of computations, the long time may not be a problem, if the computation parallelises well enough. It was mentioned in Section 5.1 that the split of $\overline{L}_n$ into the family of CSPs $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_n}$ does not provide a good parallelisation. At that point the comment rather referred to the memory usage, but the statement holds also for the runtimes of the instances. Note that each CSP can be parallelised in a naïve way by creating a separate CSP for each value of the first variable in the search order. This will in general not lead to an equal distribution of runtime, but can easily be done repeatedly. At least for a CSP with many lex-leader constraints (4.3) a case split building up on Lemma 5.1.2 will be superior to the naïve approach, since it tends to reduce the total runtime. Still, there will be CSPs with a small number of lex-leader constraints, but numerous solutions. As all of these have to be counted, splitting such a CSP into a family of CSPs will essentially have no effect on the solve time. The majority of solutions for order 9 have one or two idempotents. More specific, most solutions appear in one of two cases: on the one hand 4-nilpotent semigroups, and on the other hand semigroups which contain a 3-nilpotent subsemigroup on 9 elements. At the end of Chapter 2 it was mentioned that there is little hope to find a formula for the number of 4-nilpotent semigroups with the presented methods. How about the second case?

Question 5.6.1 Let S be a 3-nilpotent semigroup on n elements. How many non-equivalent semigroups on $n+1$ elements with S as subsemigroup exist?

Note, that if the new element is not an idempotent, the extended semigroup will be itself nilpotent. In the case that the new element is an idempotent two possibilities for the new semigroup based on S are S^0 and S^1. For the enumeration of semigroups of order 10 already an answer to a special case of Question 5.6.1 covering most such semigroups might be helpful. Much depends on whether one finds constraints that propagate well to exclude already constructed semigroups from the search.

In any case, considering the speed of hardware and software development, without doubt it will be possible to enumerate the semigroups on 10 elements using the current approach within the next 10 years.

5.6.2 Subclasses of semigroups

Some prominent subclasses of semigroups have been enumerated as well in this chapter. Separate CSPs or CSP families were created for bands and monoids. To obtain results for higher orders than presented, similar considerations as in the previous paragraphs for semigroups on 10 elements need to be made.

There are other important subclasses, which where not considered in detail; for example semigroups which are commutative, regular, or inverse.[5] The number of commutative semigroups of order 10 known from [Gri03] was confirmed by adding the constraint $T_{i,j} = T_{j,i}$ for all $i, j \in [n]$ to the CSPs in $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_{10}} \setminus \{\overline{L}_{\mathrm{id}_{10}}\}$. Solving this family of CSPs gave the expected 3 518 892 715 commutative semigroups with 10

[5]See Table 4.3 for further properties one might be interested in.

elements which are not semilattices. Details of the computation were not reported since the result is not new and no additional knowledge was used to build the CSP. For a model to be efficient it should be properly adjusted to commutativity. Similarly, it is easy to find numbers of regular and inverse semigroups of order 9 using constraints (5.12) and (5.13), see Table 5.5, but these do not efficiently use the additional structure of inverse or regular semigroups. All those subclasses are as well candidates for an extension of the data library Smallsemi. While at the moment all or none of the semigroups of a given order are contained in the library, less numerous subclasses could be included for higher orders than 8.

If one wants to go to considerably higher orders, say 15, for any particular subclass, then to get non-equivalent solutions of a CSP by posting lex-leader constraints will not be practicable for all instances. And splitting into ever larger families of CSPs is as well problematic as seen in Section 5.2. On the other hand, the number of, say inverse semigroups, is far smaller than the number of all semigroups. In this case, combining an incomplete method, which returns at least one solution from every equivalence class, with an equivalence test as post-process becomes an option. A comparison of one such approach with the lex-leader technique used throughout this thesis can be found in [DK08] – though in that instance using lex-leader was more efficient. Even better would be, of course, to prove that only a – considerably smaller – subset of lex-leader constraints has to be posted without getting equivalent solutions, or at least to find such a subset efficiently, as has been done for much simpler cases [GJMRD09].

5.6.3 Other structures and properties

Constraint satisfaction has been used to solve many different types of problems. Every finite combinatorial problem can be formulated as a CSP. There is no reason why the methods presented in this thesis would apply solely to subclasses of semigroups. Other types of algebraic objects could be enumerated, but it might not always be the most efficient approach. For example, using the CSP SL_n given in Section 5.2 was convenient to get semilattices, but it cannot compete with the orderly algorithm used in [HR02] to determine all lattices up to order 18 (equivalently, semilattices up to order 17).

For other problems, say related to graphs, an approach based on a formulation as CSP might be an option, but this is outside the immediate scope of the presented methods. The only place were graphs appeared was the construction of non-equivalent diagonals of multiplication tables. In fact, diagonals were implicitly searched for when solving L_n from CSP 4.1.3, counting all semigroups and considering diagonal positions first. Tough, it is far more efficient to calculate the set of diagonals using Algorithm 1 exploiting the correspondence to digraphs. An example of a novel result in graph theory found using constraint satisfaction is contained in [PS03]. It would be interesting to find other so far unsolved problems related to graphs, which can be solved as a CSP.

Using Minion all automorphism groups of semigroups of order at most 9 were identified. Before, it was not possible to compute the automorphism groups of semigroups with 8 elements even using most recent algorithms [ABMN09, Section 5]. This situation might seem very specific, but it indicates that constraint programming can be a useful tool for computational algebra beyond enumeration. In many situations computational algebra systems (CAS) fall back to a brute force method when looking at small problems, since elaborate algorithms used for bigger problems of the same type often produce too much overhead. This becomes a disadvantage if many simple computations have to be performed as part of a larger calculation. CSP solvers are specialised tools for the situation where search is used to solve a problem. If a solver were to be integrated in a CAS with an efficient interface certain simple computations could be performed faster.

On a more concrete note, other properties of all semigroups up to order 9 – like their Green's equivalences – could be determined using constraint satisfaction. Up to order 8 some information on Green's classes of semigroups is summarised in [SYT94]. Even for semigroups of order 8, which are now available in Smallsemi the most effective method to determine a certain property for all of them will in some cases be to modify the model $\overline{\mathcal{L}}_{\overline{\mathcal{F}}_8}$ – though Smallsemi is (hopefully) more convenient to use.

A Tables

In this first part of the appendix numbers of different types of semigroups are compiled, which did not make it to the main body of the thesis.

A.1 Nilpotent Semigroups

For the convenience of the reader this section contains the numbers of 3-nilpotent semigroups of small orders in printed form. The numbers up to equivalence and up to isomorphism were obtained using the function `Nr3NilpotentSemigroups` in Smallsemi [DM10], which is an implementation of the formulae given in Section 2.3. The numbers for all different semigroups were calculated using the formulae from Lemma 2.3.1 and Lemma 2.3.8.

The nilpotent semigroups with 2 to 9 elements are classified by rank in Table A.1. These numbers were obtained using a CSP based on the partition of nilpotent semigroups given in Lemma 2.1.2. The code implementing this approach is available in the file `nilBYrank.g` on the attached DVD (Appendix C).

Table A.1 Enumeration of non-equivalent nilpotent semigroups on $[n]$ by rank

rank $\setminus$ n	2	3	4	5	6	7	8	9
2	1	1	1	1	1	1	1	1
3		1	8	84	2 660	609 797	1 831 687 022	52 966 239 062 973
4			1	7	142	6 837	1 890 303	6 634 075 827
5				1	9	184	9 860	2 826 516
6					1	10	218	12 111
7						1	12	288
8							1	13
9								1
$\sum$	1	2	10	93	2 813	616 830	1 833 587 417	52 972 875 977 730

Table A.2 Numbers of all different, 3-nilpotent semigroups on $[n]$

n	$\lvert Z_n \rvert$
1	0
2	0
3	6
4	180
5	11 720
6	3 089 250
7	5 944 080 072
8	147 348 275 209 800
9	38 430 603 831 264 883 632
10	90 116 197 775 746 464 859 791 750
11	2 118 031 078 806 486 819 496 589 635 743 440
12	966 490 887 282 837 500 134 221 233 339 527 160 717 340
13	17 165 261 053 166 610 940 029 331 024 343 115 375 665 769 316 911 576
14	6 444 206 974 822 296 283 920 298 148 689 544 172 139 277 283 018 112 679 406 098 010
15	38 707 080 168 571 500 666 424 255 328 930 879 026 861 580 617 598 218 450 546 408 004 390 044 578 120
16	3 702 666 864 082 792 490 877 284 919 235 950 223 067 167 174 585 215 444 956 768 466 085 819 983 992 264 732 738 849 040

Table A.3 Numbers of all different, commutative, 3-nilpotent semigroups on $[n]$

n	$\lvert CZ_n \rvert$
1	0
2	0
3	6
4	84
5	1 620
6	67 170
7	7 655 424
8	2 762 847 752
9	3 177 531 099 864
10	11 942 816 968 513 350
11	170 387 990 514 807 763 280
12	11 445 734 473 992 302 207 677 404
13	3 783 741 947 416 133 941 828 688 621 484
14	5 515 869 594 360 617 154 295 309 604 962 217 274
15	33 920 023 793 863 706 955 629 537 246 610 157 737 736 800
16	961 315 883 918 211 839 933 605 601 923 922 425 713 635 603 848 080
17	160 898 868 329 022 121 111 520 489 011 089 643 697 943 356 922 368 997 915 120
18	193 723 239 221 188 065 106 566 636 546 061 286 121 446 060 611 636 162 988 630 724 403 470
19	1 471 208 650 401 311 156 222 797 795 179 822 434 242 210 115 200 377 389 766 874 742 124 806 304 222 432
20	59 648 128 317 640 935 342 568 031 970 782 068 809 723 189 235 774 109 338 822 418 510 418 694 464 097 174 910 146 340

Table A.4 Numbers of non-isomorphic 3-nilpotent semigroups on $[n]$

n	$\lvert\widehat{Z_n}\rvert$
1	0
2	0
3	1
4	9
5	118
6	4 671
7	1 199 989
8	3 661 522 792
9	105 931 872 028 455
10	24 834 563 582 168 716 305
11	53 061 406 576 514 239 124 327 751
12	2 017 720 196 187 069 550 262 596 208 732 035
13	2 756 576 827 989 210 680 367 439 732 667 802 738 773 384
14	73 919 858 836 708 511 517 426 763 179 873 538 289 329 852 786 253 510
15	29 599 937 964 452 484 359 589 007 277 447 538 854 227 891 149 791 717 673 581 110 642
16	176 968 123 463 307 372 111 212 934 463 617 568 466 247 282 881 853 775 653 620 067 093 355 211 604 330
17	18 887 344 927 589 628 463 617 942 727 388 857 157 048 886 642 440 600 912 055 301 277 528 049 508 955 221 113 399 821 320

Table A.5 Numbers of non-equivalent 3-nilpotent semigroups on $[n]$

n	$\lvert\overline{Z_n}\rvert$
1	0
2	0
3	1
4	8
5	84
6	2 660
7	609 797
8	1 831 687 022
9	52 966 239 062 973
10	12 417 282 095 522 918 811
11	26 530 703 289 252 298 687 053 072
12	1 008 860 098 093 547 692 911 901 804 990 610
13	1 378 288 413 994 605 341 053 354 105 969 660 808 031 163
14	36 959 929 418 354 255 758 713 676 933 402 538 920 157 765 946 956 889
15	14 799 968 982 226 242 179 794 503 639 146 983 952 853 044 950 740 907 666 303 436 922
16	88 484 061 731 653 686 055 606 467 231 808 786 720 624 060 411 850 795 199 165 178 781 622 833 794 930
17	9 443 672 463 794 814 231 808 971 363 694 428 578 524 510 417 979 271 827 252 285 234 996 809 337 364 454 607 870 556 069

Table A.6 Numbers of non-equivalent self-dual, 3-nilpotent semigroups on $[n]$

n	$\lvert\overline{\{S \in Z_n \mid S \text{ is self-dual}\}}\rvert$
1	0
2	0
3	1
4	7
5	50
6	649
7	19 605
8	1 851 252
9	606 097 491
10	608 877 121 317
11	1 990 358 249 778 393
12	25 835 561 207 401 249 185
13	1 739 268 479 271 518 877 288 942
14	590 686 931 539 550 985 679 107 660 268
15	846 429 051 478 198 751 690 097 659 025 763 202
16	4 975 000 837 941 847 814 744 710 290 469 890 455 985 530
17	134 193 517 942 742 449 269 192 465 569 165 773 688 102 341 290 818
18	23 019 223 063 161 156 012 642 509 011 513 682 629 696 430 762 798 444 412 800
19	30 680 509 568 610 906 776 263 624 768 434 581 568 670 347 393 800 990 316 747 809 118 850
20	252 784 530 877 599 600 371 041 762 570 280 778 746 934 981 501 931 367 532 821 102 611 466 419 351 993
21	10 755 147 520 390 108 559 858 546 036 351 594 385 615 203 989 092 989 518 572 472 023 892 431 320 586 549 067 327 578

Table A.7 Numbers of non-equivalent commutative, 3-nilpotent semigroups on $[n]$

n	$\lvert\overline{CZ_n}\rvert$
1	0
2	0
3	1
4	5
5	23
6	155
7	2 106
8	79 997
9	9 350 240
10	3 377 274 621
11	4 305 807 399 354
12	23 951 673 822 318 901
13	608 006 617 857 847 433 462
14	63 282 042 551 031 180 915 403 659
15	25 940 470 166 038 603 666 194 391 357 972
16	45 946 454 978 824 286 601 551 283 052 739 171 318
17	452 361 442 895 926 947 438 998 019 240 982 893 517 749 169
18	30 258 046 596 218 438 115 657 059 107 812 634 405 962 381 166 457 711
19	12 094 270 656 160 403 920 767 935 604 624 748 908 993 169 949 317 454 767 617 795
20	24 517 275 724 679 098 848 052 433 710 577 082 828 177 422 417 587 894 734 946 881 364 714 511
21	243 890 987 693 362 912 917 206 527 827 103 437 803 392 830 317 417 343 213 592 029 193 283 930 836 475 274
22	12 944 036 439 527 616 753 031 217 759 913 471 859 224 718 079 710 476 036 583 294 738 519 796 336 047 328 613 796 692 407

A.2 Automorphism Groups

The tables in this section list the automorphism groups of all semigroups up to equivalence with 2 to 9 elements. There is one table for each order, containing one line for each isomorphism type of automorphism group. The groups are identified by their ID in the SmallGroups library [BEO02], if their order is less than 1978. In any case a structural description, computed using the GAP command `StructureDescription`, is provided, and – of course – the number of semigroups with the given group as automorphism group.

Details about the code used to compute the automorphism groups are given in Section C.2.3.

Table A.8 Order 2

group	ID	number
trivial	$(1,1)$	3
C_2	$(2,1)$	1

Table A.9 Order 3

group	ID	number
trivial	$(1,1)$	12
C_2	$(2,1)$	5
S_3	$(6,1)$	1

Table A.10 Order 4

group	ID	number
trivial	$(1,1)$	78
C_2	$(2,1)$	39
C_2^2	$(4,2)$	3
S_3	$(6,1)$	5
S_4	$(24,12)$	1

Table A.11 Order 5

group	ID	number
trivial	$(1, 1)$	746
C_2	$(2, 1)$	342
C_3	$(3, 1)$	2
C_4	$(4, 1)$	1
C_2^2	$(4, 2)$	26
S_3	$(6, 1)$	33
D_8	$(8, 3)$	1
D_{12}	$(12, 4)$	4
S_4	$(24, 12)$	4
S_5	$(120, 34)$	1

Table A.12 Order 6

group	ID	number
trivial	$(1, 1)$	10 965
C_2	$(2, 1)$	4 121
C_3	$(3, 1)$	26
C_4	$(4, 1)$	7
C_2^2	$(4, 2)$	441
S_3	$(6, 1)$	300
D_8	$(8, 3)$	17
C_2^3	$(8, 5)$	6
D_{12}	$(12, 4)$	49
S_4	$(24, 12)$	30
S_4	$(36, 10)$	2
S_4	$(48, 48)$	4
S_5	$(120, 34)$	4
S_6	$(720, 763)$	1

Table A.13 Order 7

group	ID	number
trivial	$(1, 1)$	746 277
C_2	$(2, 1)$	76 704
C_3	$(3, 1)$	412
C_4	$(4, 1)$	82
C_2^2	$(4, 2)$	7 314
C_5	$(5, 1)$	6
S_3	$(6, 1)$	3 638
C_6	$(6, 2)$	37
$C_4 \times C_2$	$(8, 2)$	4
D_8	$(8, 3)$	169
C_2^3	$(8, 5)$	172
D_{10}	$(10, 1)$	2
D_{12}	$(12, 4)$	790
$C_2 \times D_8$	$(16, 11)$	10
S_4	$(24, 12)$	277
$C_2^2 \times S_3$	$(24, 14)$	14
S_4	$(36, 10)$	24
S_4	$(48, 48)$	45
$(S_3^2) \wr C_2$	$(72, 40)$	1
S_5	$(120, 34)$	30
$S_3 \times S_4$	$(144, 183)$	4
$C_2 \times S_5$	$(240, 189)$	4
S_6	$(720, 763)$	4
S_7	$(5040, -)$	1

Table A.14 Order 8

group	ID / order	number
trivial	$(1, 1)$	1 834 638 770
C_2	$(2, 1)$	8 176 697
C_3	$(3, 1)$	17 297
C_4	$(4, 1)$	1 270
$C_2 \times C_2$	$(4, 2)$	188 316
C_5	$(5, 1)$	92
S_3	$(6, 1)$	69 275
C_6	$(6, 2)$	1 249
$C_4 \times C_2$	$(8, 2)$	105
D_8	$(8, 3)$	2 238
$C_2 \times C_2 \times C_2$	$(8, 5)$	5 324
$C_3 \times C_3$	$(9, 2)$	5
D_{10}	$(10, 1)$	28
D_{12}	$(12, 4)$	13 583
$C_2 \times D_8$	$(16, 11)$	263
$C_2 \times C_2 \times C_2 \times C_2$	$(16, 14)$	15
$C_3 \times S_3$	$(18, 3)$	40
$C_5 : C_4$	$(20, 3)$	1
$C_4 \times S_3$	$(24, 5)$	4
S_4	$(24, 12)$	3 461
$C_2 \times A_4$	$(24, 13)$	4
$C_2 \times C_2 \times S_3$	$(24, 14)$	491
$S_3 \times S_3$	$(36, 10)$	368
$D_8 \times S_3$	$(48, 38)$	11
$C_2 \times S_4$	$(48, 48)$	768
$(S_3 \times S_3) : C_2$	$(72, 40)$	16
$C_2 \times S_3 \times S_3$	$(72, 46)$	12
$C_2 \times C_2 \times S_4$	$(96, 226)$	14
S_5	$(120, 34)$	277
$S_3 \times S_4$	$(144, 183)$	44
$\mathrm{PSL}(3, 2)$	$(168, 42)$	1
$C_2 \times S_5$	$(240, 189)$	44
$S_4 \times S_4$	$(576, 8653)$	2
S_6	$(720, 763)$	30
$S_5 \times S_3$	$(720, 767)$	4
$C_2 \times S_6$	$(1440, 5842)$	4
S_7	5040	4
S_8	40320	1

Table A.15 Order 9

group	ID / order	number
trivial	$(1, 1)$	52 961 873 376 696
C_2	$(2, 1)$	27 478 363 462
C_3	$(3, 1)$	6 329 218
C_4	$(4, 1)$	53 591
$C_2 \times C_2$	$(4, 2)$	33 882 706
C_5	$(5, 1)$	1 547
S_3	$(6, 1)$	7 881 736
C_6	$(6, 2)$	94 521
C_7	$(7, 1)$	18
$C_4 \times C_2$	$(8, 2)$	3 286
D_8	$(8, 3)$	59 125
$C_2 \times C_2 \times C_2$	$(8, 5)$	203 597
$C_3 \times C_3$	$(9, 2)$	291
D_{10}	$(10, 1)$	420
C_{10}	$(10, 2)$	108
C_{12}	$(12, 2)$	26
A_4	$(12, 3)$	3
D_{12}	$(12, 4)$	349 142
$C_6 \times C_2$	$(12, 5)$	850
D_{14}	$(14, 1)$	4
$C_4 \times C_2 \times C_2$	$(16, 10)$	18
$C_2 \times D_8$	$(16, 11)$	5 212
$C_2 \times C_2 \times C_2 \times C_2$	$(16, 14)$	1 345
$C_3 \times S_3$	$(18, 3)$	1 286
$(C_3 \times C_3) : C_2$	$(18, 4)$	1
$C_5 : C_4$	$(20, 3)$	8
D_{20}	$(20, 4)$	36
$C_7 : C_3$	$(21, 1)$	2
$C_4 \times S_3$	$(24, 5)$	105
$C_3 \times D_8$	$(24, 10)$	26
S_4	$(24, 12)$	66 700
$C_2 \times A_4$	$(24, 13)$	57
$C_2 \times C_2 \times S_3$	$(24, 14)$	14 140
$C_4 \times D_8$	$(32, 25)$	1
$(C_2 \times C_2 \times C_2 \times C_2) : C_2$	$(32, 27)$	6
$C_2 \times C_2 \times D_8$	$(32, 46)$	60
$S_3 \times S_3$	$(36, 10)$	6 188
$\mathrm{GL}(2, 3)$	$(48, 29)$	1

group	ID / order	number
$D_8 \times S_3$	$(48, 38)$	227
$C_2 \times S_4$	$(48, 48)$	12 592
$C_2 \times C_2 \times C_2 \times S_3$	$(48, 51)$	44
$(S_3 \times S_3) : C_2$	$(72, 40)$	140
$C_3 \times S_4$	$(72, 42)$	34
$C_2 \times S_3 \times S_3$	$(72, 46)$	404
$C_4 \times S_4$	$(96, 186)$	4
$C_2 \times C_2 \times S_4$	$(96, 226)$	389
S_5	$(120, 34)$	3 361
$S_3 \times S_4$	$(144, 183)$	637
$C_2 \times ((S_3 \times S_3) : C_2)$	$(144, 186)$	5
$\mathrm{PSL}(3, 2)$	$(168, 42)$	3
$D_8 \times S_4$	$(192, 1472)$	5
$S_3 \times S_3 \times S_3$	$(216, 162)$	4
$C_2 \times S_5$	$(240, 189)$	673
$C_2 \times S_3 \times S_4$	$(288, 1028)$	24
$C_2 \times C_2 \times S_5$	$(480, 1186)$	14
$S_4 \times S_4$	$(576, 8653)$	15
S_6	$(720, 763)$	259
$S_5 \times S_3$	$(720, 767)$	34
$C_2 \times S_6$	$(1440, 5842)$	31
$S_5 \times S_4$	2880	4
$S_6 \times S_3$	4320	4
S_7	5040	25
$C_2 \times S_7$	10080	4
S_8	40320	2
S_9	362880	1

A.3 Up to Isomorphism

In the main body of this thesis two semigroups are considered to have the same structure if they are isomorphic or anti-isomorphic. There are situations in which one wants to know the semigroups up to isomorphism and wants to consider semigroups that are only anti-isomorphic as different. The data library Smallsemi contains the function `UpToIsomorphism` for this purpose. Here, the most important results enumerating semigroups up to equivalence have their counterpart up to isomorphism. These are the numbers of semigroups depending on the number of idempotents in Table A.16, and the numbers of monoids in Table A.18. As con-

Table A.16 Enumeration of non-isomorphic semigroups on $[n]$ by idempotent

n / Id.	1	2	3	4	5	6	7	8	9
1	1	2	5	20	171	5 284	1 224 331	3 667 785 000	105 952 488 687 468
2		3	9	50	309	2 806	58 583	9 207 430	25 412 267 163
3			10	72	590	5 422	61 323	1 150 085	136 799 017
4				46	594	7 772	101 539	1 466 691	27 690 828
5					251	5 668	109 107	1 983 558	36 991 211
6						1 682	59 576	1 626 956	39 865 274
7							13 213	690 871	25 666 762
8								119 826	8 739 857
9									1 228 712
$\sum$	1	5	24	188	1 915	28 634	1 627 672	3 684 030 417	105 978 177 936 292

necting link (see Lemma 1.1.3) the numbers of self-dual semigroups are reported for the same cases.

Table A.17 Enumeration of non-isomorphic self-dual semigroups on $[n]$

n / Idpt.	1	2	3	4	5	6	7	8	9
1	1	2	5	18	93	930	22 899	1 937 266	613 365 656
2		1	5	24	123	754	6 721	123 988	8 265 721
3			2	16	112	764	5 567	49 969	739 317
4				6	58	542	4 751	41 939	410 158
5					19	254	2 933	31 392	328 937
6						68	1 214	17 396	223 226
7							285	6 513	113 160
8								1 376	38 979
9									7 510
$\sum$	1	3	12	64	405	3 312	44 370	2 209 839	623 492 664

Table A.18 Enumeration of non-isomorphic (self-dual) monoids on $[n]$

n	1	2	3	4	5	6	7	8	9	10
monoids	1	2	7	35	228	2 237	31 559	1 668 997	3 685 886 630	105 981 882 103 063
–, self-dual	1	2	5	19	84	509	3 901	48 957	2 264 764	625 844 421

B Semigroup Properties

This part of the appendix contains a list of standard properties of semigroups, many of which were mentioned in the thesis, but not actually used. These are in particular the properties in Table 4.3. For sake of completeness, properties which are defined in the main body of the thesis are repeated. A semigroup S is ...

... a *band* if $x^2 = x$ for all $x \in S$.

... a *Clifford semigroup* if S is regular and $ex = xe$ for all $x \in S$ and $e \in E(S)$.

... *commutative* if $xy = yx$ for all $x, y \in S$.

... *completely regular* if every element is contained in a subgroup of S.

... a *group* if S is a monoid and every element in S is invertible.

... *inverse* if for all $x \in S$ there exists a unique element $y \in S$ such that $xyx = x$.

... a *monoid* if S contains an identity.

... a *rectangular band* if S is a band and $xyx = x$ for all $x, y \in S$.

... *regular* if for all $x \in S$ there exists an element $y \in S$ such that $xyx = x$.

... *self-dual* if S is anti-isomorphic to itself.

... a *semilattice* if S is commutative and a band.

... *simple* if $SxS = S$ for all $x \in S$.

... *zero simple* if S contains a zero z and $SxS = S$ for all $x \in S \setminus \{z\}$.

C DVD Content

This last part of the appendix contains a DVD with the programs, written to obtain the computational results in this thesis. The content and how to utilise it, is described briefly. This should enable readers to repeat computations for themselves and to verify computational details of the process that led to the various enumeration results.

C.1 Smallsemi

The data library of semigroups of orders 1 to 8 is described in Section 4.2. A copy of version 0.6.0 of Smallsemi is available on the DVD. It is located in the directory `pkg/smallsemi`. This allows one to start GAP with the path to the drive containing the DVD as additional root directory (`-l` command line option) and load Smallsemi. The data files in this copy of Smallsemi are uncompressed, so that it will work on any operating system. The DVD contains in addition the two archives `smallsemi0r6p0.tar.gz` and `smallsemi0r6p0.tar.bz2`, which can be used to install Smallsemi in a Unix environment. For further information see Section 4.2 and the manual in `pkg/smallsemi/doc`.

C.2 GAP code

The implementation of Algorithm 1 and its adaptation is available in the file `construct_diagonals.g`. The functions `FunctionDigraphs` respectively `PartialFunctionDigraphs` were used to verify the numbers of non-equivalent functions and partial functions in Tables 3.1 and 3.2.

The remaining code was used to create Minion input files for the CSPs from Chapters 4 and 5 and is contained in the folder `search`. The latter has two subdirectories, `aut` and `monoids`, with the code used to obtain the results in Sections 5.5 respectively 5.4. Former versions of some of the files were written by Kelsey.

Brief descriptions of all files are given in the following. The usage of the code for the enumerative searches is explained in Section C.2.2. The functions to obtain automorphism groups look a bit different and are therefore explained separately in Subsection C.2.3.

C.2.1 Auxiliary files

Two special files are `minion.g` and `setup.g`. The former contains the interface between GAP and Minion written by Linton with slight modifications. The latter allows one to adjust search options by changing the entries of the record `FLAGS` contained in it. In particular, `FLAGS.3NIL` decides whether 3-nilpotent semigroups are counted, and `FLAGS.NTA` does the analogue for trivial action monoids in the search for monoids. An overview of the main purpose of the remaining auxiliary files is given next.

`assoc.g` creating a basic record from which Minion input is produced to search for associative multiplication tables

`blocks.g` constructing non-equivalent $\mathcal{D}$-class structures for the search of bands; based on Algorithm 2

`diagonals.g` constructing non-equivalent diagonals of multiplication tables; based on Algorithm 1

`isosymmetry.g` adding lex-leader constraints suitable to find semigroups up to isomorphism to basic record

`notselfdualsymmetry.g` adding lex-leader constraints suitable to find semigroups that are not self-dual to basic record

`selfdualsymmetry.g` adding lex-leader constraints suitable to find semigroups that are self-dual to basic record

`symmetry.g` adding lex-leader constraints suitable to find semigroups up to equivalence to basic record

The non-equivalent $\mathcal{D}$-class structures of bands were precomputed for $2 \leq n \leq 10$ and stored in the files `bounds`n`.g`.

C.2.2 Enumeration of semigroups

Table C.1 contains an overview of the files providing code to enumerate different types of semigroups using various case splits to build a family of CSPs. The table serves as well as reference stating which code was used for which of the enumeration results. The description of the auxiliary files can be found in the previous section.

One special case is the file `basic.g` (in the directory `search`), which provides the code to search for semigroups depending on the diagonal, and in addition the function `nrSemigroups`, which uses all refinements described in Section 5.3.1. A function `getSemigroups` to obtain all multiplication tables of semigroups is provided as well, though if one just wants the result, it is quicker to use the function `RecoverMultiplicationTable` from Smallsemi.

To use any of the code to enumerate semigroups, GAP has to be started from within the directory `search`. Moreover, an executable for Minion is needed, which is assumed to be called by the command `minion`. The default setting can be overwritten in the file `setup.g`, or the variable `MINION_EXEC` can be adjusted within GAP. Note that for some of the bigger computations more than 4 GB RAM – and consequently 64-bit executables – are required.

All files for the enumerative search listed in Table C.1 contain usually only one function called `search...`, where the suffix can depend on the type and on the case split. Calling that function without arguments will print advice on how to use it. An example for the search of semigroups using the case split on the number of idempotents looks as follows.

```
[andreas@kininvie search]$ gap idempotent.g
[...]
gap> searchByIdempotent(4);
Semigroups of order 4
Idempotents 1 of 4 with 12 symmetries and 19 solution(s)
Idempotents 2 of 4 with 8 symmetries and 37 solution(s)
Idempotents 3 of 4 with 12 symmetries and 44 solution(s)
Idempotents 4 of 4 with 48 symmetries and 26 solution(s)
GAP cpu    : 0:00:00.284
Minion cpu : 0:00:00.060
126
gap>
```

Table C.1 Overview of code for the enumeration of semigroups

file	type of semigroup	case split on	used for	command
`search/`				
`all.g`	all different semigroups	–	Table 4.1	`search_ALL_Semigroups`
`allones.g`	constant diagonal	first row	`basic.g`	`searchAllOnes`
`bandsBYrect.g`	bands	minimal $\mathcal{D}$-class	Table 5.3	`searchBands`
`bands.g`	bands	$\mathcal{D}$-class structure	Table 5.3	`searchBands`
`basic.g`[a]	semigroups	diagonal	Table 5.1	`searchByDiagonal`
`comminv.g`	comm. inverse	number of idempotents	Table 5.5	`searchCommInverseSemigroups`
`commutative.g`	commutative	diagonal	Table 5.5	`searchAbelianByDiagonal`
`idempotent.g`	semigroups	number of idempotents	Table 5.2	`searchByIdempotent`
`inverse.g`	inverse	number of idempotents	Table 5.5	`searchInverseSemigroups`
`nilpotent.g`	nilpotent	number of annihilators	`basic.g`	`searchNilpotentSemigroups`
`nilBYrank.g`	nilpotent by rank	partition from Lemma 2.1.2	Table A.1	`searchNilpotentByRank`
`regular.g`	regular	number of idempotents	Table 5.5	`searchRegularSemigroups`
`semilattices.g`	semilattices	–	Table 5.3	`searchSemilattices`
`single.g`	semigroups	–	Table 4.2	`searchAllSemigroups`
`search/monoids/`				
`basic.g`	monoids	diagonal	Table 5.6	`searchMonoidsByDiagonal`
`groupdiags.g`	monoids	unit group and diagonal	Table 5.7	`searchMonoidsByGroupAndDiag`
`group.g`	monoids	unit group	Table 5.7	`searchMonoidsByGroup`
`single.g`	monoids	–	Table 5.6	`searchAllMonoids`

[a] See text for additional functions in `basic.g`

C.2.3 Computing automorphism groups

The folder `search/aut` contains files to control the lex-leader constraints (4.3) according to the different cases depending on the type of automorphism group as described in Section 5.5.

`C2symmetry.g` requires one automorphism of order 2; no other orders.

`V4symmetry.g` requires three automorphisms of order 2; no other orders.

`C2-3symmetry.g` requires seven automorphisms of order 2; no other orders.

`C2+4symmetry.g` requires at least eight automorphisms of order 2; no other orders.

`evensymmetry.g` requires at least one automorphism of order 4; none whose order is not a power of 2.

`oddsymmetry.g` requires at least one automorphism whose order is an odd prime.

In addition, a copy for some of the files from Section C.2.2 is stored in `aut`. The difference to the original files is, that instead of numbers of semigroups, lists of Booleans corresponding to their automorphisms are returned. This allows to reconstruct the automorphism group in GAP and identify it using the function `IdSmallGroup`.

Again, GAP has to be started from within the directory `search`. Reading the file `aut/getAutomorphismGroups.g` will load a function `getAutos`. This function takes $n \in \mathbb{N}$ as input and returns a list of pairs, each containing a group ID and the number of semigroups of order n with the specified group as automorphism group. If the order of the group is greater than 2010, then an ID is not available and a string describing the structure of the group is given instead. The latter is computed with `StructureDescription`.

For 3-nilpotent semigroups of orders 8 and 9 the automorphism groups are precomputed and stored in the files `autos-3nil-8.txt` and `autos-3nil-9.txt` respectively. For these values of n one can set `FLAGS.3NIL` to `false`. Then the automorphism groups of 3-nilpotent semigroups are not determined by `getAutos`, but the precomputed information is used instead.

C.3 Instances and Output

To give a complete record of the computations undertaken to obtain the presented enumeration results, copies of the instances and output files are included on the DVD. They are located in the directories `instances` and `output` respectively. An overview of the subdirectories is given in Table C.2.

The instances were run with the script `runscript.sh` in each folder, and the results were extracted from the output with the script `stats.sh`. The original versions of both scripts were written by Kelsey. Neither of the scripts will work on the DVD itself, since the directory has to be writable. The instances have been compressed with gzip [Deu96] to fit on the DVD. They can be run without decompressing them if gzip is available.

Table C.2 Overview of directories containing Minion instances and output files

subdirectory	CSP (family)	max. order	results in
semigroups			
`ALL`	L_n	7	Table 4.1
`basic`	$\overline{\mathcal{L}}_{\overline{\mathcal{F}}_n}$	8	Table 5.1
`basic-3nil`	$\overline{\mathcal{L}}^{-3}_{\overline{\mathcal{F}}_n}$	8	Table 5.1
`idempotents-3nil`	$\overline{\mathcal{L}}^{-3}_{\overline{\mathcal{U}}}$	8	Table 5.2
`idempotents`	$\overline{\mathcal{L}}_{\overline{\mathcal{U}}}$	8	Table 5.2
`single`	$\overline{L}_n$	8	Table 4.2
`single-3nil`	$\overline{L}_n^{-3}$	8	Table 4.2
bands			
`bandsBYrect`	$\overline{\mathcal{L}}_{\overline{\mathcal{R}}_n}$	9	Table 5.3
`bands-single`	$\overline{L}_{\mathrm{id}_n}$	8	Table 5.3
`semilattices`	$\overline{SL}_n$	10	Table 5.3
monoids			
`monoids-single`	$\overline{M}_n$	9	Table 5.6
`monoidsBYdiagonal`	$\overline{\mathcal{M}}_{\overline{\mathcal{P}}_{n-1}}$	9	Table 5.6
`monoidsBYdiagonalE`	$\overline{\mathcal{M}}^{e}_{\overline{\mathcal{P}}_{n-1}}$	9	Table 5.6
`monoidsBYgroupE`	$\overline{\mathcal{M}}^{e}_{\overline{\mathcal{G}}_{\leq n}}$	10	Table 5.7
`monoidsBYgroupNTA`	$\overline{\mathcal{M}}^{nta}_{\overline{\mathcal{G}}_{\leq n}}$	10	Table 5.7
`monoidsBYgroupANDdiag`	$\overline{\mathcal{M}}_{\overline{\mathcal{G}}_{\leq n},\overline{\mathcal{F}}^*_{\leq n}}$	9	Table 5.7
`monoidsBYgroupANDdiagE`	$\overline{\mathcal{M}}^{e}_{\overline{\mathcal{G}}_{\leq n},\overline{\mathcal{F}}^*_{\leq n}}$	10	Table 5.7
`monoidsBYgroupANDdiagNTA`	$\overline{\mathcal{M}}^{nta}_{\overline{\mathcal{G}}_{\leq n},\overline{\mathcal{F}}^*_{\leq n}}$	10	Table 5.7

Bibliography

[ABE+03] E. Aichinger, F. Binder, J. Ecker, P. Mayr, and C. Nöbauer.
SONATA – system of near-rings and their applications.
`http://www.algebra.uni-linz.ac.at/Sonata/`, 2003.
A GAP 4 package [GAP08], Version 2.

[ABMN09] J. Araújo, P. V. Bünau, J. D. Mitchell, and M. Neunhöffer.
Computing automorphisms of semigroups.
Journal of Symbolic Computation, In Press, Corrected Proof:–, 2009.
`http://www.sciencedirect.com/science/article/B6WM7-4XJP3XV-1/2/fe6e8fcf4f3ab95f67dc757f58891b39`.

[BEO02] Ulrich Besche, Bettina Eick, and Eamonn O'Brien.
The SmallGroups Library.
`http://www-public.tu-bs.de:8080/~beick/soft/small/small.html`, 2002.
An accepted GAP 4 package [GAP08].

[CGLR96] James M. Crawford, Matthew L. Ginsberg, Eugene M. Luks, and Amitabha Roy.
Symmetry-breaking predicates for search problems.
In Luigia Carlucci Aiello, Jon Doyle, and Stuart Shapiro, editors, *KR'96: Principles of Knowledge Representation and Reasoning*, pages 148–159, San Francisco, California, 1996. Morgan Kaufmann.

[CP61] A. H. Clifford and G. B. Preston.
The algebraic theory of semigroups. Vol. I.
Mathematical Surveys, No. 7. American Mathematical Society, Providence, R.I., 1961.

[Dav53] Robert L. Davis.
The number of structures of finite relations.
Proc. Amer. Math. Soc., 4:486–495, 1953.

[dB59] N. G. de Bruijn.
Generalization of Polya's fundamental theorem in enumerative combinatorial analysis.

Nederl. Akad. Wetensch. Proc. Ser. A 62 = Indag. Math., 21:59–69, 1959.

[Deu96] P. Deutsch.
GZIP file format specification version 4.3.
Alassin Enterprises, ftp://ftp.isi.edu/in-notes/rfc1952.txt, 1996.

[DK08] Andreas Distler and Tom Kelsey.
The monoids of order eight and nine.
In S. Autexier, J. Campbell, J. Rubio, V. Sorge, M. Suzuki, and F. Wiedijk, editors, *Artificial Intelligence and Symbolic Computation, 8th International Conference, AISC 2008, Birmingham, July, 2004, Proceedings*, volume 5144 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2008.

[DK09] Andreas Distler and Tom Kelsey.
The monoids of orders eight, nine & ten.
Ann. Math. Artif. Intell., 56(1):3–21, 2009.

[DM10] Andreas Distler and James D. Mitchell.
Smallsemi - A library of small semigroups.
http://www-history.mcs.st-and.ac.uk/~jamesm/smallsemi/, Feb 2010.
A GAP 4 package [GAP08], Version 0.6.0.

[ER63] P. Erdős and A. Rényi.
Asymmetric graphs.
Acta Math. Acad. Sci. Hungar, 14:295–315, 1963.

[For55] George E. Forsythe.
SWAC computes 126 distinct semigroups of order 4.
Proc. Amer. Math. Soc., 6:443–447, 1955.

[For60] George E. Forsythe.
Review of: 'On Finite Semigroups' by John L. Selfridge.
Mathematics of Computation, 14(70):204–207, 1960.
http://www.jstor.org/stable/2003217.

[GAP08] The GAP Group, (http://www.gap-system.org).
GAP – Groups, Algorithms, and Programming, Version 4.4.12, 2008.

[GJM06] Ian P. Gent, Christopher Jefferson, and Ian Miguel.
Minion: A fast scalable constraint solver.
In Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, editors, *The European Conference on Artificial Intelligence 2006 (ECAI 06)*, pages 98–102. IOS Press, 2006.

[GJMRD09] A. Grayland, C. Jefferson, I. Miguel, and C. M. Roney-Dougal.

Minimal ordering constraints for some families of variable symmetries.
Ann. Math. Artif. Intell., 57(1):75–102, 2009.

[Gri95] P.-A. Grillet.
Semigroups - An introduction to the structure theory, volume 193 of *Monographs and Textbooks in Pure and Applied Mathematics*.
Marcel Dekker Inc., New York, 1995.

[Gri03] Pierre Antoine Grillet.
Computing finite commutative semigroups. II, III.
Semigroup Forum, 67(2):159–184, 185–204, 2003.

[Gri07] Pierre Antoine Grillet.
Computing finite semigroups. I. The first row.
Semigroup Forum, 74(1):41–54, 2007.

[Har66] Michael A. Harrison.
The number of isomorphism types of finite algebras.
Proc. Amer. Math. Soc., 17:731–737, 1966.

[How95] John M. Howie.
Fundamentals of semigroup theory, volume 12 of *London Mathematical Society Monographs. New Series.*
The Clarendon Press Oxford University Press, New York, 1995.
Oxford Science Publications.

[HP73] Frank Harary and Edgar M. Palmer.
Graphical enumeration.
Academic Press, New York, 1973.

[HR02] Jobst Heitzig and Jürgen Reinhold.
Counting finite lattices.
Algebra Universalis, 48(1):43–53, 2002.

[JMS91] H. Jürgensen, F. Migliorini, and J. Szép.
Semigroups.
Akadémiai Kiadó (Publishing House of the Hungarian Academy of Sciences), Budapest, 1991.

[Jür89] H. Jürgensen.
Annotated tables of semigroups of orders 2 to 6.
Technical Report TR-231, Department of Computer Science, The University of Western Ontario, 1989.

[Jür78] H. Jürgensen.
Computers in semigroups.
Semigroup Forum, 15(1):1–20, 1977/78.

[JW77] H. Jürgensen and P. Wick.

Die Halbgruppen der Ordnungen ≤ 7.
Semigroup Forum, 14(1):69–79, 1977.

[KJ56] V. L. Klee Jr.
The November meeting in Los Angeles.
Bull. Amer. Math. Soc., 62(1):13–23, 1956.

[Kos82] A. I. Kostrikin.
Introduction to algebra.
Springer-Verlag, New York, 1982.
Translated from the Russian by Neal Koblitz, Universitext.

[KR85] Václav Koubek and Vojtěch Rödl.
Note on the number of monoids of order n.
Comment. Math. Univ. Carolin., 26(2):309–314, 1985.

[KRS76] Daniel J. Kleitman, Bruce R. Rothschild, and Joel H. Spencer.
The number of semigroups of order n.
Proc. Amer. Math. Soc., 55(1):227–232, 1976.

[Leo91] Jeffrey S. Leon.
Permutation group algorithms based on partitions. I. Theory and algorithms.
J. Symbolic Comput., 12(4-5):533–583, 1991.
Computational group theory, Part 2.

[LR04] Eugene M. Luks and Amitabha Roy.
The complexity of symmetry-breaking formulas.
Ann. Math. Artif. Intell., 41(1):19–45, 2004.

[MS55] T. S. Motzkin and J. L. Selfridge.
Semigroups of order five.
presented in [KJ56], 1955.

[Mut88] L. R. Mutafchiev.
Limit theorem concerning random mapping patterns.
Combinatorica, 8(4):345–356, 1988.

[Nöb97] Christof Nöbauer.
GLISSANDO: Small Semigroups and Nearrings.
`http://www.gap-system.org/Gap3/Packages3/gliss.html`, 1997.
A GAP 3 share package [S$^+$97], Version 1.0.

[Neu79] Peter M. Neumann.
A lemma that is not Burnside's.
Math. Sci., 4(2):133–141, 1979.

[Ple67] Robert J. Plemmons.
There are 15973 semigroups of order 6.
Math. Algorithms, 2:2–17, 1967.

[Ple69] Robert J. Plemmons.
A survey of computer applications to semigroups and related structures.
SIGSAM Bull., 12:28–39, 1969.

[Ple70] Robert Plemmons.
Construction and analysis of non-equivalent finite semigroups.
In *Computational Problems in Abstract Algebra (Proc. Conf., Oxford, 1967)*, pages 223–228. Pergamon, Oxford, 1970.

[Pol37] G. Polya.
Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen.
Acta Math., 68(1):145–254, 1937.

[PS03] Karen E. Petrie and Barbara M. Smith.
Symmetry breaking in graceful graphs.
In *Principles and Practice of Constraint Programming – CP 2003, LNCS 2833*, pages 930–934. Springer, 2003.

[Rea78] Ronald C. Read.
Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations.
Ann. Discrete Math., 2:107–120, 1978.
Algorithmic aspects of combinatorics (Conf., Vancouver Island, B.C., 1976).

[Red27] J. Howard Redfield.
The Theory of Group-Reduced Distributions.
Amer. J. Math., 49(3):433–455, 1927.

[Rio60] J. Riordan.
The enumeration of trees by height and diameter.
IBM J. Res. Develop., 4:473–478, 1960.

[RvBW06] Francesca Rossi, Peter van Beek, and Toby Walsh.
Handbook of Constraint Programming (Foundations of Artificial Intelligence).
Elsevier Science Inc., New York, NY, USA, 2006.

[S+97] Martin Schönert et al.
GAP – Groups, Algorithms, and Programming – version 3 release 4 patchlevel 4.
Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Germany, 1997.

[Slo09] N. J. A. Sloane.
The on-line encyclopedia of integer sequences.

`http://www.research.att.com/~njas/sequences/Seis.html`, 2009.

[SYT94] S. Satoh, K. Yama, and M. Tokizawa. Semigroups of order 8. *Semigroup Forum*, 49(1):7–29, 1994.

[Tam53] Takayuki Tamura. Some remarks on semi-groups and all types of semi-groups of order $2, 3$. *J. Gakugei Tokushima Univ.*, 3:1–11, 1953.

[Tam54] Takayuki Tamura. Notes on finite semigroups and determination of semigroups of order 4. *J. Gakugei. Tokushima Univ. Math.*, 5:17–27, 1954.

[Tam73] Dov Tamari. The associativity problem for monoids and the word problem for semigroups and groups. In *Word problems: decision problems and the Burnside problem in group theory (Conf., Univ. California, Irvine, Calif., 1969; dedicated to Hanna Neumann)*, pages 591–607. Studies in Logic and the Foundations of Math., Vol. 71. North-Holland, Amsterdam, 1973.

[THA+55] Kazutoshi Tetsuya, Takao Hashimoto, Tadao Akazawa, Ryōichi Shibata, Tadashi Inui, and Takayuki Tamura. All semigroups of order at most 5. *J. Gakugei Tokushima Univ. Nat. Sci. Math.*, 6:19–39. Errata on loose, unpaginated sheet, 1955.

[WWL81] S. K. Winker, L. Wos, and E. L. Lusk. Semigroups, antiautomorphisms, and involutions: a computer solution to an open problem. I. *Math. Comp.*, 37(156):533–545, 1981.