On Algorithmic Properties Of Computable Magmas

by Trang Ha

M.A. in Mathematics, May 2015, The George Washington University

A Dissertation submitted to

The Faculty of
The Columbian College of Arts and Sciences
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Summer August 31, 2018

Dissertation directed by

Valentina Harizanov
Professor of Mathematics

The Columbian College of Arts and Sciences of The George Washington University certifies that Trang Ha has passed the Final Examination for the degree of Doctor of Philosophy as of May 14^{th}, 2018. This is the final and approved form of the dissertation.

On Algorithmic Properties Of Computable Magmas

Trang Ha

Dissertation Research Committee:

Director: Valentina Harizanov, Professor of Mathematics

Reader: Jennifer Chubb, Associate Professor of Mathematics,
University of San Francisco

Reader: Józef Przytycki, Professor of Mathematics

Reader: Alexander Shumakovitch, Assistant Professor of Mathematics

Examiners:

Departmental examiner: Yongwu Rong, Professor of Mathematics

Outside examiner: Hosam Mahmoud, Professor of Statistics, GWU

Committee Chair: Joel Lewis, Assistant Professor of Mathematics

Acknowledgments

First, I would like to express my greatest gratitude to my advisor, Valentina Harizanov, who has been a wonderful mentor to me since my first day of graduate school. Valentina has taught me so many invaluable lessons about logic, research, and the worlds inside and outside of academia. She has helped me to stay on track and overcome any obstacles along the way. I would not be where I am today without her guidance and constant support for the past six years.

This dissertation would have never been complete without the assistance of our visiting scholar, Jennifer Chubb. I feel extremely lucky that you visited GW in my last year of graduate school. Thank you for spending an immense amount of your time and energy working with me, reading my thesis, and providing insightful comments. Your encouragement after each of our meetings motivated me to keep pushing and finish my thesis.

I would also like to thank my friends and family for always being there for me. Many thanks go to my best friend, Dicko Sow, who never stops believing in me. Your continued support and our long phone calls have helped me to stay optimistic about both school and life. Lots of thanks to my boyfriend, Aaron Long, for being the consistent positive force in my life. Thanks for bringing me so much laughter while making sure that I would never quit my program. I am also grateful that you introduced me to running and hiking. Short runs and wonderful memories of being surrounded by nature have helped to keep me sane through stressful moments. Most importantly, I would not have achieved higher education without the tremendous support from my parents. Not being able to attend college themselves, they sacrificed and worked endless hours to ensure that my sister and I always have access to any level of education that we wanted. Thank you for investing so much in us and our education.

Last but not least, I am thankful for all of my professors and teachers who have shown me the beauty of mathematics and science.

Abstract

Using the notions and methods of computability theory, we study effective properties of computable magmas. A magma is an algebraic structure with a single binary operation that is not necessarily associative nor commutative. We define a magma to be computable if it is finite or if its domain can be identified with the set of natural numbers and the magma operation is computable. Our main focus is on order relations on magmas, which provide a ranking of the elements of the magma while staying invariant under the magma operation. The algorithmic complexity of additional relations, which are not part of the basic structure, such as orderings of the domain, might change under isomorphic transformations of the structure even if the structure remains computable.

We formulate sufficient and necessary algebraic conditions for a binary relation to be an order on a given magma. Using these conditions, we construct a binary tree to represent the orders on magmas in an informative and visual way. The tree construction allows us to formulate a necessary and sufficient condition for extending a partial order to a total order. A tree representation of a magma's orderings in an effective setting facilitates analysis of computability-theoretic complexity of the order

relations. We also study geometric properties of the space of orders on a magma using the natural topology defined on binary trees (i.e., Cantor space). We investigate the Turing degrees of left orders, right orders, and bi-orders of orderable magmas from various algebraic classes.

We give special attention to classes of self-distributive magmas that come from knot theory, known as racks and quandles. At present, very little is known about their orderability and orders. We consider specific instances of quandles, such as the conjugate quandles of groups. We construct an isomorphic computable copy of the conjugate quandle of the free group with infinitely many generators so that the quandle has no computable right orders. As a corollary, we obtain that the space of right orders on this quandle is homeomorphic to the Cantor set. The same is true for its space of bi-orders.

Finally, we study global decision problems for magmas, that is, we ask if there is an algorithm capable of deciding whether an arbitrary computable magma satisfies some specified property. When there is no such algorithm, we ask how hard it is to detect the property in a computability-theoretic sense. Equivalently, we discuss the complexity of index sets of magmas that satisfy certain properties within the class of computable magmas. We provide sharp characterizations of the algorithmic complexity of detecting many natural properties of magmas in terms of the arithmetical hierarchy. Properties considered include commutativity, idempotence, right self-distributivity, orderability, and left-inverse property.

Contents

Chapter 1

Introduction to Computability Theory

1.1 Basic Notions

Computability theory (also called recursion theory) is a branch of mathematical logic where we study algorithms and decision problems. One of the first famous decision problems was proposed by Hilbert in 1928 and is known as the *Entscheidungsproblem*. The problem is to find an algorithm that will decide whether a given first-order mathematical sentence is valid or not. It was clear that the solution to this problem would require a formal definition of *algorithm*. The search for the answer to *Entscheidungsproblem* led to the rise of computability theory in the following decades. The area flourished in the 1930s when mathematicians Gödel, Church, Turing, Kleene, Post, and others started formulating fundamental basic notions and proving foundational results.

In 1936, two prominent candidates for a definition of algorithms were independently proposed by mathematicians Alonzo Church and Alan Turing. Church invented the $\lambda-$*calculus*, a formal logic system that can be used to define functions by encoding the natural numbers that are now referred to as the *Church numerals*. In the same year, Turing introduced a precise notion of abstract computation devices now known as *Turing machines*, which formalized the intuitive notion of an algorithm. These machines are capable of following a list of instructions and executing calculations on a given input by reading and updating symbols on an infinite tape in an automatic way; the list of instructions for the symbols is called a *Turing program*. Church's and Turing's definitions of an algorithm turned out to be equivalent: they both capture the essence of what computable functions should be. Such a function should be given as a procedure that takes in an input, runs a series of calculations on this input, and returns a unique output if the process terminates. This allows us to provide a formal definition of computable functions as follows:

Definition 1.1.1. A function $f : \mathbb{N} \to \mathbb{N}$ is *computable* if there is a Turing program P that computes f. That is, given an input $x \in \mathbb{N}$, P is run on x, the computation ends in a finite number of steps and produces $f(x)$ as the output.

Although the definition above seems to only mention functions that are defined on $\mathbb{N}$, we can actually use any countable sets that can be identified with the natural numbers in a systematic way, for instance, by Gödel codes, to be the domain and codomain of f since the Gödel coding method allows us to encode elements of these countable sets using natural numbers. This is why in computability theory, for convenience, we usually consider $\mathbb{N}$ as the domain and codomain of functions. Moreover, if f is only defined on a subset of $\mathbb{N}$, we call f a *partial computable function*, i.e., there is some Turing program P where $P(x)$ returns output y if and only if x is in the domain of f and $f(x) = y$. On the other hand, if the domain of f is the entire set of natural numbers, f is a *total computable function*.

Following the definition of computable functions above, it is not too hard to see that basic, elementary functions, such as addition and exponential functions, are computable; the Turing machines in these cases are fairly easy to construct. However, when descriptions of functions get complicated, it is much more difficult to construct a Turing program to calculate the functions. The *Church – Turing thesis* addresses this matter. The thesis asserts that if a function is effectively calculable in the intuitive sense, then it can be computed via some Turing machine. Although there can be no formal proof, there is not, as of yet, any evidence that would refute the validity of the thesis, and it is widely accepted in the mathematical world.

As each Turing program contains a finite list of instructions, which themselves are finite strings on a finite alphabet, there are only countably many Turing programs. We can sort these programs via the shortlex ordering, where we utilize the lexicographical order and the length of each string. Therefore, there is an algorithmic way to enumerate all partial computable functions. In other words, given $k \geq 1$, we can obtain a list of all k-ary partial computable functions:

$$\varphi_0^{(k)}, \varphi_1^{(k)}, \ldots, \varphi_n^{(k)}, \ldots$$

where we have an algorithmic procedure to connect an index i with a Turing machine that computes $\varphi_i^{(k)}$. Thus, each $\varphi_i^{(k)}$ is identified with the Turing machine M_i. When $k = 1$ we omit the superscript. Furthermore, using a diagonal argument, we can show that it is *not* possible to enumerate *only* total functions in this list in an algorithmic way.

Once we establish the definition of a computable function, we can define other computability-theoretic concepts, such as computable sets and computable relations. A set (or a relation) is computable if there is some computer program that can decide if a given element belongs to the set (or we can determine if the relation holds true for a given tuple of elements). More formal definitions are given via the characteristic functions below.

A set $A \subseteq \mathbb{N}$ is *computable* if its characteristic function χ_A, with

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A \end{cases}$$

is a computable function.

In a similar fashion, a relation R is *computable* if its characteristic function χ_R, with

$$\chi_R(\vec{x}) = \begin{cases} 1 \text{ if } R(\vec{x}) \text{ holds,} \\ 0 \text{ if } R(\vec{x}) \text{ does not hold} \end{cases}$$

is computable.

Most of the functions that we can think of are computable as we intuitively tend to describe these objects in an algorithmic way. However, since we know there are uncountably many functions on $\mathbb{N}$, and only countably many computable functions, most functions out there are *not* computable. A well-known example of a non–

computable set was described by Alan Turing's in 1936. The halting set, denoted by K, is defined as

$$K = \{e : e \in dom(\varphi_e)\}$$

So, K consists of all indices e where the eth Turing program halts on input e. The interesting point here is that there is a possibility $\varphi_e(e)$ will run forever, and the non–computability of K means that there is no algorithm capable of determining whether $\varphi_e(e)$ will halt in a finite number of steps.

Using a diagonal argument, Turing showed that the halting set is not computable, and a negative answer to the *Entscheidungsproblem* follows. Despite being non–computable, K can be algorithmically listed out, which leads us to another fundamental notion in computability theory called *computable enumerability.*

Definition 1.1.2. A set $A \subseteq \mathbb{N}$ is *computably enumerable* (c.e.) if its partial characteristic function $\psi_A(x)$, where

$$\psi_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ \uparrow & \text{if } x \notin A \end{cases}$$

is a partial computable function. We use the up-arrow, $\uparrow$, to indicate that the algorithm fails to halt.

A relation is *computably enumerable* if it is a c.e. subset of $\mathbb{N}^k$ for some natural number k.

There are equivalent characterizations of c.e. sets that are often useful, for example, a set A is c.e. if and only if it is the domain of some partial computable function, or equivalently, if there is a total computable function g that enumerates it, that is,

A is the range of some total computable function. The halting set K is an example of a set that is c.e. but not computable.

1.2 Relative Computability

Perhaps the single most important concept in computability theory is that of classifying the complexity of sets and the problems they may encode based on *relative computability*, a notion that Turing introduced in his later work [44]. We allow programs to access an oracle set $\mathcal{O} \subseteq \mathbb{N}$ during the computation. This extra, possibly non–algorithmic information allows *relative* computation of more complicated sets or problems. The oracle set supplies additional information on demand by disclosing its elements and non–elements when queried by the Turing program. Note that as it was the case for the standard partial computable functions, we can algorithmically enumerate all partial X-computable oracle functions:

$$\varphi_0^X, \varphi_1^X, \ldots, \varphi_n^X, \ldots$$

Definition 1.2.1. Let $A, B \subseteq \mathbb{N}$. A set A is called *B-computable* if A can be computed by an oracle Turing machine with oracle B. In other words, there exists an oracle program φ_e^B such that for each element x in A, $\chi_A(x) = \varphi_e^B(x)$ and the domain of φ_e^B is $\mathbb{N}$.

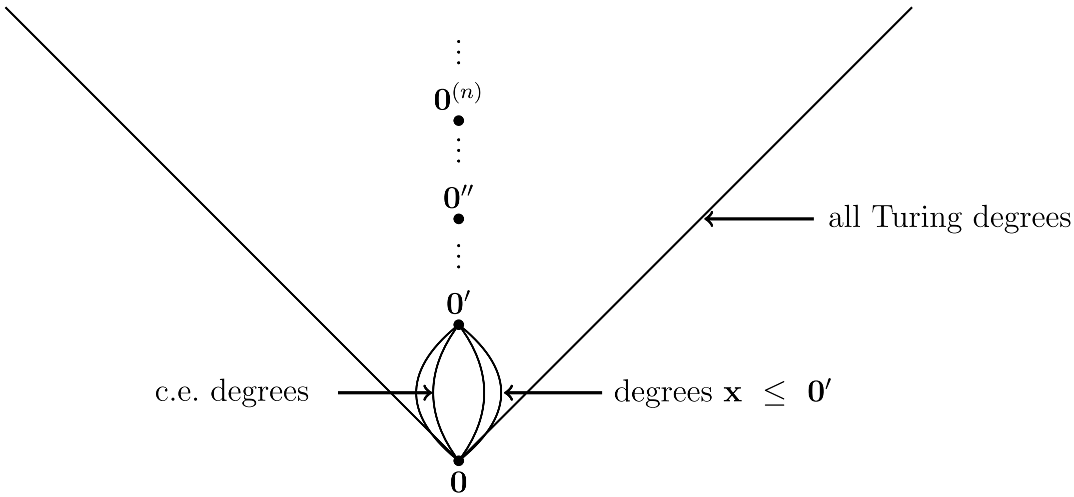
We also say that A is *Turing-reducible* to B, or that A is *computable relative* to B, and write $A \leq_T B$ or $deg(A) \leq \deg(B)$.

Relative computability provides us with a way to compare sets of natural numbers and the information they encode. To be more specific, the Turing reducible relation divides subsets of $\mathbb{N}$ into equivalence classes called *Turing degrees*. We say that sets A and B have the same *Turing degree* if and only if A is B-computable and B is A-computable, i.e., A and B encode the same information. We write $deg(A) = deg(B)$, or $A \equiv_T B$ where $\equiv_T$ stands for Turing equivalence of sets. For example, any set and its complement have the same Turing degree. Since there are $2^{\aleph_0}$ subsets of natural numbers and only countably infinite many Turing oracle programs, there are $2^{\aleph_0}$ Turing degrees.

The first Turing jump of a set X is defined as $X' = \{n : n \in dom(\varphi_n^X)\}$. We denote the Turing degree of X by $\mathbf{x} = \deg(X)$, and the Turing degree of X' by $\mathbf{x}'$. In particular, $\mathbf{0}'$ denotes the Turing degree of the halting set K, and thus, the halting set is also called the *first Turing jump* of the empty set (i.e., set of computable sets).

It is not hard to show that $\mathbf{x} < \mathbf{x}'$; this implies that each jump generates properly more information than the base degree. Moreover, we can iterate Turing jump to higher levels. The nth Turing jump of the empty set is $\emptyset^{(n)}$, and let $\mathbf{0}^{(n)} = \deg(\emptyset^{(n)})$. We will denote the set of all Turing degrees by $\mathcal{D}$. Then $\mathcal{D}$ is partially ordered by $\leq_T$, but it is not totally ordered because one can construct sets that are Turing incomparable [9]. Moreover, $\mathcal{D}$ is closed under supremum; indeed the join of two degrees is $\mathbf{a} \vee \mathbf{b} = \deg(A \oplus B)$, where $A \in \mathbf{a}$ and $B \in \mathbf{b}$. However, it has been shown that there are incomparable c.e. degrees with no infimum, so $\mathcal{D}$ is not closed under infimum. Therefore, the partially ordered set $\mathcal{D}$ forms an upper semilattice. This semilattice is vast and it already gets very complicated for degrees of computably enumerable sets. We visualize the diagram of Turing degree hierarchy under the jump operator in Figure 1.1.

FIGURE 1.1: Turing degrees $(\mathcal{D}, \leq)$



Degrees $\mathbf{x} \leq \mathbf{0}'$ is also called Δ^0_2 degrees, which will be introduced in the next section.

1.3 The Arithmetical Hierarchy

Another way to measure the complexity of sets is to consider the complexity of their defining formulas. We use blocks of alternating quantifiers to formulate a statement that represents a given set. Moreover, the formula must always be written in its *prenex normal form*, where all quantifiers are stacked together in the front and are followed by a quantifier-free formula.

Definition 1.3.1. Let A be a subset of $\mathbb{N}$.

(i) A is Σ^0_n if A can be expressed as

$$A = \{y \in \mathbb{N} : (\exists x_1)(\forall x_2)(\exists x_3) \dots (Qx_n)[R(\vec{x}, y)]\}$$

where Q is either $\exists$ or $\forall$ depending on the parity of n, and R is a computable relation.

(ii) A is Π^0_n if A can be expressed as

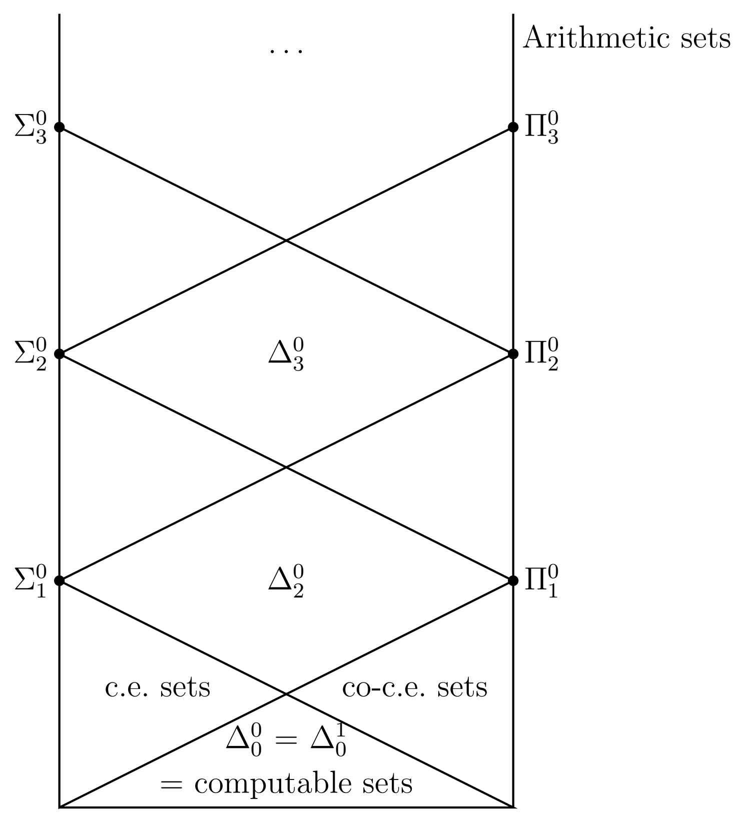
$$A = \{y \in \mathbb{N} : (\forall x_1)(\exists x_2)(\forall x_3)\dots(Qx_n)[R(\vec{x}, y)]\}$$

where Q is either $\exists$ or $\forall$ depending on the value of n, and R is a computable relation.

(iii) A is Δ^0_n if A is both Σ^0_n and Π^0_n.

The superscript "0" indicates that the set is defined by a formula in the language of first-order arithmetic, so the scope of quantifiers is $\mathbb{N}$. These formulated definitions give rise to the *arithmetical hierarchy*, which is visualized in Figure 1.2 below [41, 40].

FIGURE 1.2: The Arithmetical Hierarchy



A Venn diagram of the arithmetical hierarchy. Note that the diagram does not represent a partition in the hierarchy of $\mathcal{P}(\mathbb{N})$, and Σ^0_n, Π^0_n, Δ^0_n are closed downwards under $\leq$.

The arithmetical hierarchy is sometimes also called the *Kleene-Mostowski hierarchy* since it was originally developed independently by Kleene [26] and Mostowski [35].

A set is *arithmetic* if it belongs to Σ^0_n for some $n \in \mathbb{N}$. As both Turing degrees and the arithmetical hierarchy contain information about the complexity of a given set, one expects to observe a strong connection between the arithmetical hierarchy and Turing complexity. Post described this relationship in the following theorem.

Theorem 1.3.2 (Post's Theorem)**.** *For any $A \subseteq \mathbb{N}$ and $n \leq 0$, we have*

(i) $A \in \Sigma^0_{n+1} \iff A$ *is c.e. in some* Π^0_n *set* $\iff A$ *is c.e. in some* Σ^0_n *set;*

(ii) $\emptyset^{(n)}$ *is* Σ_n*-complete for* $n > 0$*;*

(iii) $A \in \Sigma^0_{n+1} \iff A$ *is c.e. in* $\emptyset^{(n)}$*;*

(iv) $A \in \Delta^0_{n+1} \iff A \leq_T \emptyset^{(n)}$*.*

One example of an application of Post's Theorem lies at the lowest level of the hierarchy: a set A is computably enumerable if and only if A is Σ^0_1. For more on computability-theoretic notions, we recommend [41].

1.4 Computable Structure Theory

We take a *structure* or a *model* for a language $\mathcal{L}$ to be a mathematical object that consists of a nonempty set of elements (called the *domain* or the *universe* of the structure) together with functions, relations, and constants associated to symbols with appear in $\mathcal{L}$. For instance, the language of arithmetic is $\mathcal{L}_A = \{0, 1, +, \cdot, \leq\}$, and the standard model of arithmetic is composed of the set of all natural numbers with the usual binary operations of addition and multiplication, constants 0 and 1, and relations $\leq$.

In this dissertation, we focus on first-order theories and models, where formulas are built using constants, symbols for relations and functions, variables (standing for the elements of the domain), Boolean operation symbols, and universal and existential quantifiers ranging over first-order variables. First-order languages are sufficient to express a great deal of everyday mathematics.

A *sentence* is defined to be a formula with no free variables (that is, all variables fall in the scope of a quantifier), and a *theory* is any set of sentences. Every model has its (full) theory. For example, full number theory is the set of all first-order sentences true in the standard model of arithmetic. We define the *atomic diagram* of a structure $\mathcal{A}$ to be the set of all atomic sentences and negations of atomic sentences in the language of the structure augmented with new constant symbols for all elements of the domain.

Definition 1.4.1. Let $\mathcal{A}$ be an $\mathcal{L}$ - structure for a computable language $\mathcal{L}$. We say $\mathcal{A}$ is *computable* if its domain is a computable set and its atomic diagram is computable as a set under some suitable Gödel encoding. Equivalently, $\mathcal{A}$ is computable if its domain is computable, and all functions and relations in $\mathcal{A}$ are uniformly computable.

Computable structure theory has been an active research area; its main objective concerns the algorithmic aspects and the computability theoretic properties of mathematical structures. Researchers in the area have been studying *how computable* certain mathematical objects can be with or without the presence of additional information. Ash and Knight compiled a great monograph for this subject in [2] where they provide definability results on various structures and general techniques for building and studying computable models. Another excellent reference for this topic can be found at [19].

Chapter 2

Orderings on Magmas

2.1 Motivation

Orderable algebraic structures play an important role in several areas of mathematics. They have been studied extensively by algebraists [27, 36, 28, 20] and more recently by topologists and computability theorists [43, 38, 39, 42, 11, 12, 8]. All surface groups, with the exception of the projective plane and the Klein bottle groups, are bi-orderable. All knot groups are left-orderable and many are bi-orderable. Surprisingly many fundamental groups of 3-manifolds are left-orderable or even bi-orderable.

In this chapter, we apply the techniques of computability theory to further analyze the spaces of orders on orderable magmas. A *magma* M is a non-empty set with a binary operation $\cdot : M \times M \to M$. A countable magma $(M, \cdot)$ is *computable* if its domain M is a computable set and its operation $\cdot$ is computable. For any infinite computable magma we may assume, without loss of generality, that its domain is the set of all natural numbers.

A binary relation R on M is a *right order* on $(M, \cdot)$ if R is a strict linear ordering (i.e., an irreflexive total ordering) of the domain M and is right-invariant under the magma operation:

$$(\forall a, b, c \in M)[(a, b) \in R \Rightarrow (a \cdot c, b \cdot c) \in R].$$

The notion of *left order* is defined similarly. A binary relation R on M is a *bi-order* on $(M, \cdot)$ if R is a linear ordering of the domain M and

$$(\forall a, b, c \in M)[(a, b) \in R \Rightarrow ((a \cdot c, b \cdot c) \in R \wedge (c \cdot a, c \cdot b) \in R)].$$

By $RO(M)$ we denote the set of all right orders on $(M, \cdot)$, by $LO(M)$ the set of all left orders, and by $BiO(M)$ the set of all bi-orders on $(M, \cdot)$. Clearly, $BiO(M) = RO(M) \cap LO(M)$.

For a group G, every left order $<_l$ on G induces an associated right order $<_r$ on G defined by

$$a <_r b \Leftrightarrow b^{-1} <_l a^{-1}.$$

This is because if $a <_r b$, then for any c in G, $ac <_r bc$. By definition, $(bc)^{-1} <_l (ac)^{-1}$, which is equivalent to $c^{-1}b^{-1} <_l c^{-1}a^{-1}$. As c is arbitrary, this implies that $b^{-1} <_l a^{-1}$ and $<_l$ is a left order on G. Of course, the reverse direction of the argument holds as well.

It is easy to see that if G is a left-orderable group, then G must be *torsion-free*, i.e., the identity element is the only torsion element in the group G. The converse is however, not true — there are torsion-free groups that are not left-orderable. One example is the crystallographic group

$$\Gamma = \langle a, b : a^2ba^2 = b, b^2ab^2 = a \rangle.$$

This group is shown to be torsion-free by looking at the 4-Klein group, a non-Abelian group that is isomorphic to a quotient of Γ. Furthermore, due to the group's relations, for any ε, δ in $\{1, -1\}$, we get $(a^\varepsilon b^\delta)^2(b^\delta a^\varepsilon)^2 = i_\Gamma$, where i_Γ is the identity element of Γ. Γ fails to be left-orderable because under an arbitrary left order relation, regardless of the choices of signs for a and b, $(a^\varepsilon b^\delta)^2(b^\delta a^\varepsilon)^2$ will either be positive or negative, which is a contradiction. More detailed explanation can be found at [15].

Nevertheless, groups from many important classes, such as torsion-free abelian groups or more generally, torsion-free nilpotent groups, are bi-orderable. Recall that a group is *nilpotent* if it has a lower central series that terminates with the trivial group after finitely many steps. Nilpotent groups are known as "almost Abelian" in the sense that the only nilpotent group of class 0 is the trivial group and nilpotent groups of class 1 consists of all non-trivial abelian groups.

There are also left-orderable groups that are not bi-orderable – an example of such a group is the fundamental group of the Klein bottle

$$K = \langle a, b \mid aba^{-1} = b^{-1} \rangle.$$

As this group has an isomorphic one-relator group presentation, it is locally indicable, and thus left-orderable. It can be shown to be not bi-orderable via a contradiction argument: Assume that $<$ is a bi-order on K. Then either $b > e$ or $b < e$, where e is the identity of K. If $b > e$, by the presentation of K, $b^{-1} = aba^{-1} > e$. On the other hand, if $b < e$, then $b^{-1} = aba^{-1} < e$. In either case, we end up with a contradiction; hence, K is not bi-orderable.

Linnell [29] showed that a left-orderable group has either finitely or uncountably many left orders. On the other hand, there is a bi-orderable group with countably infinitely many bi-orders (see [4]).

2.2 Ordering Properties of Magmas

Here we give algebraic conditions for a binary relation to be an ordering on an arbitrary magma.

Let X be a nonempty set and R, S be binary relations on X. Define the inverse relation of R, $R^{-1} \subseteq X \times X$, as

$$R^{-1} = \{(b, a) \mid (a, b) \in R\},$$

and the composition $S \circ R \subseteq X \times X$ as

$$S \circ R = \{(x, z) : (\exists y)[(x, y) \in R \wedge (y, z) \in S]\}.$$

The diagonal of X, Δ_X, is defined by

$$\Delta_X = \{(a, a) : a \in X\}.$$

Let $(M_1, *_1), (M_2, *_2)$ be magmas. The direct product of these magmas, which we write as $(M_1, *_1) \times (M_2, *_2)$, is defined as one might expect. The underlying set is the Cartesian product $M_1 \times M_2$, and the binary operation $*$ is defined component-wise:

$$(a, b) * (c, d) = (a *_1 c, b *_2 d).$$

Let $(Q, *)$ be a magma and S, T be subsets of Q. The *product of subsets S and T* is defined by

$$ST = \{s * t : s \in S \wedge t \in T\}.$$

We can now describe algebraic conditions that suffice for a binary relation to be a left order on an arbitrary magma.

Proposition 2.2.1. *Let $\mathcal{M} = (M, *)$ be a magma. A binary relation R on $\mathcal{M}$ is a left order on $\mathcal{M}$ if and only if R satisfies all of the four following properties:*

1. $R \circ R \subseteq R$,
2. $R \cap R^{-1} = \emptyset$,
3. $R \cup R^{-1} = (M \times M) - \Delta_M$,
4. $\Delta_{M} R \subseteq R$.

Proof. Property 1 guarantees transitivity of R, Property 2 shows anti-symmetry and Property 3 ensures the totality of R. Property 4, where the product is in $(M, *) \times (M, *)$, guarantees that R is left-invariant under the magma operation.

On the other hand, if R is a left order on the magma $(M, *)$, R satisfies all ordering axioms; thus, the four properties above follows.

□

It is easy to see that there are modifications of Property 4 to ensure right or bi orderings on $(M, *)$. For a right order R on $(M, *)$, instead of 4, we require

4′. $R\Delta_{M} \subseteq R$,

and for a bi-order R on $(M, *)$, instead of 4, we need

4″. $(\Delta_{M} R \cup R\Delta_{M}) \subseteq R$.

These algebraic conditions of an ordering on a magma are consistent with the notion of ordering conditions for a group, which have been extensively studied [42, 43]. As each group is equipped with an identity and inverse elements, for convenience, a partial left order $\prec$ on a group G is determined by and often identified with its *positive partial cone*:

$$P = \{a \in G : e \preceq a\},$$

where $e \in G$ is the identity element. Similarly, the *negative partial cone* is defined as

$$P^{-1} = \{a \in G : a \preceq e\}.$$

We can easily verify that P is a *subsemigroup* of G (i.e., $PP \subseteq P$), which is *pure* (i.e., $P \cap P^{-1} = \{e\}$). Such a subsemigroup $P \subseteq G$ defines a left order on G if and only if P is *total* (i.e., $P \cup P^{-1} = G$). P defines a bi-order on G if, in addition, P is a *normal* subsemigroup of G (i.e., $g^{-1}Pg \subseteq P$ for every $g \in G$). For more details of how the positive cone has been used to study the space of orders on groups, the author recommends [28].

2.3 Spaces of Orders on Computable Magmas

In this section, we endow the set of left, right, and bi-orders of a magma with a natural topology and show that the corresponding space of orders are effectively closed subspaces of Cantor space, 2^ω. Then, we transfer certain computability-theoretic results about effectively closed sets to spaces of orders on magmas.

We define a topology on $LO(M)$ by choosing as a subbasis the collection

$$\mathcal{S} = \{S_{(a,b)}\}_{(a,b)\in(M\times M)-\Delta_M},$$

where $S_{(a,b)} = \{R \in LO(M) | \ (a,b) \in R\}$ and $\Delta_M = \{(a,a) \mid a \in M\}$. A basic fact from topology tells us that a topological space is zero-dimensional if it is a T_1-space with a clopen basis. Since S clearly has these properties, Proposition 2.3.1 is immediate.

Proposition 2.3.1. *The space $LO(M)$ is zero-dimensional.*

Using Vedenissoff's theorem about embedding a zero-dimensional space into the Cantor cube, Dabkowska, et al. proved that if a magma M has cardinality $\mathfrak{m} \geq \aleph_0$, the space of left orderings, $LO(M)$, is a compact space that can be embedded into the $\mathfrak{m}$-dimension Cantor cube $\{0,1\}^{\mathfrak{m}}$ [11]. Similarly, it follows that $RO(M)$ is compact. By definition, the space $BiO(M)$ inherits the same topology from $LO(M)$, thus, it too is a closed subspace of $LO(M)$ and also compact. Moreover, since $LO(M)$ is embeddable into the Cantor cube, the only totally disconnected perfect compact metric space, if M is countable, then by Urysohn's metrization theorem, $LO(M)$ is metrizable.

The topology defined above has been shown to be useful in the study of spaces of orderings of groups. For a computable torsion-free abelian group of rank 1, the space of bi-orders has exactly two elements. Sikora [39] established that for finite $n \geq 2$, the space $BiO(\mathbb{Z}^n)$ is homeomorphic to the Cantor set. Let $\mathbb{Z}^\omega$ be $\bigoplus_{i\in\omega}\mathbb{Z}$, the direct sum of ω copies of $\mathbb{Z}$. Dabkowska [10] proved that the space $BiO(\mathbb{Z}^\omega)$ is homeomorphic to the Cantor set. Later, Chubb [7] obtained this result as a corollary of a result in [17].

Computability-theoretic properties such as orderings are not necessarily preserved under isomorphisms; however, they are preserved under computable isomorphisms. Hence, any computable copy of $(\mathbb{Z}, +)$ has exactly two orders (the usual one and its inverse) and they are both computable. Moreover, if G is a computable group, then any right order $\prec_r$ and the induced left order $\prec_l$ have the same Turing degree. Solomon [42] proved that for $n \geq 2$, a computable group isomorphic to $(\mathbb{Z}^n, +)$ has an order in every Turing degree. On the other hand, Downey and Kurtz [17] constructed a computable group isomorphic to $(\mathbb{Z}^\omega, +)$ with no computable order. Dobritsa [16] showed that every computable, torsion-free, abelian group is isomorphic to a computable group with a computable order. Recently, Harrison-Trainor [22] proved that there is a computable left-orderable group that is not isomorphic to a computable group with a computable left order. It is not known whether this is true for the case of bi-orderable groups. Solomon also proved that a computable, torsion-free, abelian group G of infinite rank has an order in every Turing degree $\mathbf{x} \geq \mathbf{0}'$, and that a computable, torsion-free, properly n-step nilpotent group has an order in every Turing degree $\mathbf{x} \geq \mathbf{0}^{(n)}$. Note that these recent results about the space of orderings of computable groups are also relevant to the study of the space of orderings of computable magmas. All existential theorems are true for magmas as a group is a magma. Furthermore, the work that has been done for orderings of groups can be used to generate ideas for investigating orderings of magmas.

2.3.1 Trees of Orderings

A subtree $\mathcal{S}$ of the full binary tree is a subset of $2^{<\omega}$ which is closed under initial segments. Such a tree $\mathcal{S}$ is computable if its set of nodes and the ordering are computable. A subset of the Cantor set 2^{ω} is called *effectively closed* if it is the collection of infinite paths through a computable subtree of $2^{<\omega}$. Note that there is a computable infinite binary tree without a computable infinite path.

Metakides and Nerode [34] investigated trees and orderable fields. A field is orderable exactly when it is *formally real*, which means if any sum of squares of elements in the field equals 0, then each of those elements must be equal to 0. [1] Metakides and Nerode showed that the sets of all orders on computable formally real fields are in exact correspondence to the effectively closed subsets of 2^{ω}. Solomon [43] established that for every orderable computable group G, there is a computable binary branching tree $\mathcal{T}$ and a Turing degree preserving bijection from $BiO(G)$ to the set of all infinite paths of $\mathcal{T}$. Chubb [7] has a similar result for computable orderable semigroups. Here, we present such a result for *any* magma in general and investigate how certain computability-theoretic results for effectively closed sets can be transferred to orders on computable magmas.

Theorem 2.3.2. *Let $(M,\cdot)$ be an infinite right-orderable computable magma. Then there is a computable binary tree $\mathcal{T}_M$ such that the right orders on $(M,\cdot)$ exactly correspond to infinite paths in $\mathcal{T}_M$, and the Turing degree is preserved via a bijection*

[1] There are other equivalent properties that the field can satisfy to become formally real; for instance, a field is formally real if -1 is not a sum of squares in the field.

between the right orders on $(M, \cdot)$ and the infinite paths of $\mathcal{T}_M$.

Proof. Without loss of generality, we can assume that the domain M is the set of all natural numbers. Let $p_0, p_1, p_2, \ldots$ be a computable enumeration of all elements of $(M \times M) - \Delta_M$. For $p = (a, b)$, let $p^* = (b, a)$. We use characteristic functions of sets of pairs of elements of M to describe the infinite paths in the tree $\mathcal{T}$. We will show that there is a bijection between the right orders on $(M, \cdot)$ and the infinite paths of $\mathcal{T}$, and that the Turing degree is preserved.

For a right order R on $(M, \cdot)$, define $f_R \in 2^\omega$ by

$$f_R(i) = \begin{cases} 1 & \text{if } R(a,b) \wedge (p_i = (a,b)), \\ 0 & \text{if } R(b,a) \wedge (p_i = (a,b)). \end{cases}$$

It is easy to see that R and f_R have the same Turing degree. For every $f \in 2^\omega$, define

$$R_f = \{p_i : f(i) = 1\} \cup \{p_i^* : f(i) = 0\}.$$

Then f and R_f have the same Turing degree. Clearly, $R = R_{f_R}$ and $f = f_{R_f}$.

We will build $\mathcal{T}$ in stages. At every stage $s \geq 0$, we will have a finite tree $\mathcal{T}_s$ that will consist of all nodes of $\mathcal{T}$ of length $\leq s$. We will have $\mathcal{T}_s \subseteq \mathcal{T}_{s+1}$ and $\mathcal{T} = \bigcup_{s \geq 0} \mathcal{T}_s$. For each $\sigma \in \mathcal{T}$, we define a finite label set S_σ, where $S_\sigma \subseteq M \times M$, which can be viewed as possibly determining a partial order on $(M, \cdot)$. For $f \in 2^\omega$ and $n \in \mathbb{N}$, by $f \upharpoonright n$ we denote $(f(0), \ldots, f(n))$.

Construction:

Stage 0: Set $\mathcal{T}_0 = \{\langle\ \rangle\}$, the tree with the empty sequence $\langle\ \rangle$ (root), as its only member. Set $S_{\langle\ \rangle} = \emptyset$.

Stage $s+1$: Consider each node σ of length s of $\mathcal{T}_s$. If S_σ contains (a,b) and (b,a) for any $a \neq b$, then that node is declared to be terminal, so it will not be further extended. Otherwise, extend σ by adding both $\sigma^\frown 0$ and $\sigma^\frown 1$ to $\mathcal{T}_{s+1}$. Add p_s to $S_{\sigma^\frown 1}$, and add p_s^* to $S_{\sigma^\frown 0}$. Furthermore, for $i = 0, 1$:

1. add (s,s) to $S_{\sigma^\frown i}$,
2. add every new (a,c) to $S_{\sigma^\frown i}$ such that for some b, both (a,b) and (b,c) are in S_σ,
3. add all new (ac, bc) to $S_{\sigma^\frown i}$ for every (a,b) in S_σ and every $c \leq s$.

End of the construction.

It follows from the construction that $\mathcal{T}$ is computable. The following lemmas complete the proof.

Lemma 2.3.3. *Let f be an infinite path in $\mathcal{T}$. Then R_f is a right order on $(M, \cdot)$.*

Proof. First we show that R_f is total. For every a, we have $(a,a) \notin R_f$ by the definition of the sequence $p_0, p_1, p_2, \ldots$. Assume that a, b are elements in M such that $a \neq b$. There are $i, j \in \mathbb{N}$ such that $p_i = (a,b)$ and $p_j = p_i^*$. Then $f(i) = 1 - f(j)$ since no initial segment of f terminates at any stage. Hence, exactly one of $(a,b) \in R_f$, $(b,a) \in R_f$ holds.

Now, assume that $(a,b), (b,c) \in R_f$. Let i, k be such that $p_i = (a,b)$ and $p_k = (b,c)$. Then $f(i) = 1$ and $f(k) = 1$, so $(a,b) \in S_{f\restriction(i+1)}$ and $(b,c) \in S_{f\restriction(k+1)}$. Let $s = \max\{i+1, k+1\}$. Then, we have $(a,c) \in S_{f\restriction(s+1)}$. Let $p_m = (a,c)$. Then, we have $f(m) = 1$ and hence, $(a, c) \in R_f$. Therefore, R_f satisfies the transitive property of order relations.

Lastly, we show that R_f is right invariant. Let $(a, b) \in R_f$ and $c \in M$. Let i be such that $p_i = (a, b)$. Then $f(i) = 1$, so $(a, b) \in S_{f\restriction(i+1)}$. Let $s = \max\{i+1, c\}$. Then $(a \cdot c, b \cdot c) \in S_{f\restriction(s+1)}$. Let $p_\ell = (a \cdot c, b \cdot c)$. Then $f(\ell) = 1$ since no initial segment of f terminates at stage $\ell + 1$. Hence, $(a \cdot c, b \cdot c) \in R_f$.

As it satisfies all axioms of a right order, R_f is a right order on $(M, \cdot)$.

□

Lemma 2.3.4. *Let R be a right order of $(M, \cdot)$. Then f_R is an infinite path in $\mathcal{T}$.*

Proof. By induction on $n \geq 0$, we prove that for every n,

$$(f_R \restriction n) \in \mathcal{T} \wedge (S_{f_R\restriction n} - \Delta_M) \subseteq R.$$

Let $n = 0$ (note that this base case corresponds to *Stage* 1 in the construction). We have $(f_R \restriction 0) \in \mathcal{T}$ since $f_R \restriction 0 = f_R(0)$ and $(0), (1) \in \mathcal{T}$. Let $p_0 = (a, b)$. If $(a, b) \in R$, then $f_R(0) = 1$, and $S_{f_R\restriction 0} = \{(a, b), (0, 0)\}$, so $(S_{f_R\restriction 0} - \Delta_M) = (a, b) \subseteq R$. On the other hand, if $(b, a) \in R$, then $f_R(0) = 0$, and $S_{f_R\restriction 0} = \{(b, a), (0, 0)\}$, so $(S_{f_R\restriction 0} - \Delta_M) = (b, a) \subseteq R$.

Assume that the statement holds for n. Let $\sigma = f_R \restriction n$. Then, since $(S_\sigma - \Delta_M) \subseteq R$ and R is a strict linear ordering of M, there is no pair a, b such that $a \neq b$ and $(a, b), (b, a) \in S_\sigma$. By our construction, both $\sigma^\frown 0$ and $\sigma^\frown 1$ are included in $\mathcal{T}$ at stage $n+1$, so $(f_R \restriction (n+1)) \in \mathcal{T}$. Either p_n or p_n^* belongs to R, so we can assume, without loss of generality, that $p_n \in R$. Then $f_R(n + 1) = 1$ and $p_n \in S_{f_R\restriction(n+1)}$. Since R is closed under transitive closure and multiplication on the right by an element of M and S is extended by preserving its transitivity and right invariant multiplication via steps 2 and 3 in the construction, we have $(S_{f_R\upharpoonright(n+1)} - \Delta_M) \subseteq R$.

Our argument above show that our statement holds for $n+1$ and thus, it is true for all n. Therefore, f_R is an infinite path in $\mathcal{T}$.

□

Now, we define a computable functional Ψ from the set of all right orders on $(M, \cdot)$ to the set of all infinite paths of $\mathcal{T}$ by $\Psi(R) = f_R$. Clearly, $\Psi^{-1}(f) = R_f$ and Ψ is a Turing-degree preserving computable bijection.

□

2.3.2 Applications

The theorem above tells us that for each computable copy of a computable right-orderable magma, the tree of orderings that we construct is computable. Hence, by the Low Basis Theorem of Jockusch and Soare [23], there exists a right order of low Turing degree for any of these magmas. Recall that a set X and its Turing degree $\mathbf{x}$ are *low* if $X' \leq_T \emptyset'$, hence $\mathbf{x}' = \mathbf{0}'$. Moreover, it can be shown that for every left-orderable (bi-orderable, respectively) computable magma $(M, \cdot)$, there is a computable binary tree $\mathcal{T}$ and a Turing degree preserving bijection from $LO(M)$ ($BiO(M)$, respectively) to the set of all infinite paths of $\mathcal{T}$. Therefore, a computable left-orderable (bi-orderable, respectively) magma has a left order (bi-order, respectively) of low Turing degree.

Since the topology of the space of orders does not change under isomorphism, it is an invariant of the isomorphism class. In the next three propositions, we use "order" to mean any of a right order, a left order, or a bi-order.

It is not hard to see that an isolated path in a computable binary tree must be computable. Furthermore, as we know that a non-empty closed subspace of a Cantor space that has no isolated point is perfect, a computable binary tree with infinite paths and no computable ones must have that the space of its paths is homeomorphic to the Cantor set. Applying this knowledge to Theorem 2.3.2, we obtain the following proposition.

Proposition 2.3.5. *Let $(M, \cdot)$ be an orderable magma.*

(i) If $(M, \cdot)$ is computable and it has only finitely many orders, then they must be all computable.

(ii) If $(M, \cdot)$ is computable and it has countably infinitely many orders, then it has infinitely many computable orders.

(iii) If $(M, \cdot)$ has a computable copy which does not have a computable order, then the space of orders on $(M, \cdot)$ is homeomorphic to the Cantor set.

By its definition, the set of all infinite paths through a computable binary branching tree is a Π^0_1 class. Equivalently, we can view a Π^0_1 class as an effectively closed subset of ω^ω. Therefore, using Theorem 2.3.2, we can apply results about effectively closed sets (or Π^0_1 class) to obtain further complexity results about spaces of orders on magmas. The following proposition is acquired by implementing the well-known low basis theorem, the hyperimmune-free basis theorem, and a couple results about

Turing degrees of certain Π^0_1 classes that were established by Jockusch and Soare in [23]. Recall that a set A is *hyperimmune* if and only if it is infinite and there is no disjoint strong array[2] of finite sets where A has non-empty intersection with each of them. A degree $\mathbf{a}$ is hyperimmune-free if no hyperimmune set has degree $\mathbf{a}$. (For a more thorough discussion of a hyperimmune-free Turing degrees, see [41] or [40]).

Proposition 2.3.6. *Let $(M, \cdot)$ be an infinite computable orderable magma.*

(i) Then $(M, \cdot)$ has an order of low *Turing degree.*

(ii) Then $(M, \cdot)$ has an order of hyperimmune-free *Turing degree.*

(iii) Assume that $(M, \cdot)$ does not have a computable order. Let $(\mathbf{d}_i)_{i \in I}$ be any countable sequence of non-computable (i.e., non-zero) Turing degrees. Then $(M, \cdot)$ has $2^{\aleph_0}$ orders of mutually incomparable Turing degrees such that they are also incomparable with each $\mathbf{d}_i$ for $i \in I$.

(iv) Then $(M, \cdot)$ contains two orders of Turing degrees $\mathbf{a}$ and $\mathbf{b}$ such that the infimum of $\mathbf{a}$ and $\mathbf{b}$ is the computable degree, $\mathbf{0}$.

The next proposition about orders comes from applying Theorem 2.3.2 to the c.e. basis theorem and some other results by Jockusch and Soare about effectively closed sets which does not consist of computable elements [24]. We recall that a Turing degree is computably enumerable if it contains a computably enumerable set, and computably enumerable degrees form a countably infinite upper semilattice.

Proposition 2.3.7. *Let $(M, \cdot)$ be a computable orderable magma.*

[2] A sequence $\{p_i\}_{i \in \omega}$ of finite sets is defined to be *a strong array* if there is a computable function f such that for every $i \in \omega$, we have $p_i = D_{f(i)}$, where $(D_x)_{x \in \omega}$ is some canonical numbering of the finite sets, and all $D_{f(i)}$'s are pairwise disjoint.

(i) Then $(M, \cdot)$ *has an order of a computably enumerable Turing degree.*

(ii) If $(M, \cdot)$ *does not have a computable order, then there is a computably enumerable Turing degree* $\mathbf{a}$ *such that* $(M, \cdot)$ *has no order of Turing degree* $\leq \mathbf{a}$.

2.3.3 Strong Degrees

We extend our discussion to a refinement of the Turing degrees known as the *truth-table* degrees. Recall that $(\varphi_n^X)_{n\geq 0}$ is a computable enumeration of all unary partial computable functions with oracle X. A set A is *truth-table reducible* to a set B, in symbols $A \leq_{\mathrm{tt}} B$, if there is a computable function h and an index n such that for every x,

$$A(x) = \varphi_n^{B\upharpoonright h(x)}(x),$$

and for any string $\sigma \in 2^{<\omega}$ of length $h(x)$, we have $x \in dom(\varphi_n^\sigma)$. Here, as is usual in computability theory, the computation $\varphi_n^\sigma(x)$ assumes that only questions about numbers in the domain of σ are posed to the oracle. Nonetheless, the algorithm here is much more robust in the sense that it must halt on every input even when the oracle does not produce correct information about B.

We say that a set $X \leq_T \emptyset'$ is *super low* if $X' \leq_{\mathrm{tt}} \emptyset'$. Hence, by the *Super Low Basis Theorem* of Jockusch and Soare (see [40]), a computable right-orderable (left-orderable, bi-orderable, respectively) magma contains a *super low* right order (left order, bi-order, respectively).

2.4 Extending Partial Orders on Magmas

Orders are often obtained by extending partial orders. A binary relation $\prec$ on a magma $(M, \cdot)$ is a *partial left order* on $(M, \cdot)$ if $\prec$ is a partial ordering on the domain M and

$$(\forall a, b, c \in M)[a \prec b \Rightarrow (c \cdot a \text{ and } c \cdot b \text{ are comparable}) \wedge (c \cdot a \prec c \cdot b)].$$

Not all partial orders can be extended to total orders. In the following proposition, we give a condition that is necessary and sufficient to guarantee that we *can* extend a given partial order to a total order.

Proposition 2.4.1. *Let $(M, \cdot)$ be a magma and $R \subseteq M \times M$ be some partial order on $(M, \cdot)$. Then M admits a total order extending R if and only if for every finite tuple $\vec{p}$, where $\vec{p} = \{p_0, p_1, p_2, \dots, p_n\} \subset (M \times M) - \Delta_M$, there is a corresponding sequence $\vec{\epsilon} = (\epsilon_0, \epsilon_1, \epsilon_2, \dots, \epsilon_n)$, where $\epsilon_i \in \{+1, -1\}, i = 1, 2, \dots, n$, such that for any (x, y) in $(M \times M) - \Delta_M$, it is not the case that both (x, y) and (y, x) are in $(R \cup \vec{p}^{\,\vec{\epsilon}})$.* $(*)$

Proof. ($\implies$) Suppose we have R to be a partial order on $(M, \cdot)$ that can be extended to a total order S. We need to find $\vec{\epsilon}$ such that $((x, y) \wedge (y, x)) \notin (R \cup \vec{p}^{\,\vec{\epsilon}})$ for any $(x, y) \in (M \times M) - \Delta_M$.

Since S is a total order, based on the properties of orders on magmas that we discussed in section **2.2**, for each $i = 1, 2, \dots, n$, either p_i or $p_i^{-1} \in S$. If $p_i \in S$, then we choose $\epsilon_i = 1$; otherwise, let $\epsilon_i = -1$. Applying this to the entire tuple $\vec{p}$, we will end up with the desired sequence $\vec{\epsilon}$ of elements from $\{+1, -1\}$.

($\Longleftarrow$) Before we prove the other direction of the proposition, we will prove the following claim.

Claim 2.4.2. *If R satisfies the condition $(*)$, then for all $p \in (M \times M) - \Delta_M$, either R extended by p or R extended by p^{-1} satisfies the condition $(*)$.*

We use $R \cup p$ to denote R extended by p, and $R \cup p^{-1}$ to denote R extended by p^{-1}.

Proof. We prove the claim by contradiction.

Suppose that neither $R \cup p$ nor $R \cup p^{-1}$ satisfies $(*)$. Then there exist some finite tuples $\overrightarrow{q}, \overrightarrow{r}$ of elements from $(M \times M) - \Delta_M$, such that for all sequences $\overrightarrow{\epsilon}$ and $\overrightarrow{\delta}$ of elements from $\{+1, -1\}$, we can find some $(a, b) \in (M \times M) - \Delta_M$ where $((a, b) \wedge (b, a)) \in (R \cup \overrightarrow{q}^{\overrightarrow{\epsilon}})$ and some $(c, d) \in (M \times M) - \Delta_M$, where $((c, d) \wedge (d, c)) \in (R \cup \overrightarrow{r}^{\overrightarrow{\delta}})$.

Denote $\overrightarrow{\epsilon^\star} = \{1, \overrightarrow{\epsilon}\}$, $\overrightarrow{\delta^\star} = \{-1, \overrightarrow{\delta}\}$, $\overrightarrow{q^\star} = p \cup \overrightarrow{q}$, and $\overrightarrow{r^\star} = p^{-1} \cup \overrightarrow{r}$. With this new notation, we have $((a, b) \wedge (b, a)) \in (R \cup \overrightarrow{q^\star}^{\overrightarrow{\epsilon^\star}})$ and $((c, d) \wedge (d, c)) \in (R \cup \overrightarrow{r^\star}^{\overrightarrow{\delta^\star}})$.

Thus, we have just found finite tuples $\overrightarrow{q^\star}$ and $\overrightarrow{r^\star}$ with their corresponding sequences $\overrightarrow{\epsilon^\star}$ and $\overrightarrow{\delta^\star}$ of elements from $\{+1, -1\}$ where the extension of R to these tuples contains both some pair of elements in $M \times M$ and its reverse pair. This contradicts the fact that R satisfies the condition $(*)$. Therefore, by contradiction, either R extended by p or R extended by p^{-1} satisfies the condition $(*)$.

$\square$

Now we go back to proving our proposition. Let S be the maximal partial order extending R for which S satisfies $(*)$. We will show by contradiction that S is, in fact, a total order.

Suppose there exists $p \in (M \times M) - \Delta_M$, such that both p and $p^{-1} \notin S$. Since S satisfies $(*)$, by the claim we proved above, either $R \cup p$ or $R \cup p^{-1}$ satisfies $(*)$.

However, since $S \cup p$ and $S \cup p^{-1}$ are extensions of R, and R is a subset of both of these extensions, this contradicts that S is a maximal partial order. Thus, S must be total.

□

Taking the partial order R in Proposition 2.4.1 to be $\emptyset$, we obtain the following corollary that gives a necessary and sufficient condition for a magma to be orderable.

Corollary 2.4.3. *A magma M is orderable if and only if for every finite tuple $\overrightarrow{p}$ where $\overrightarrow{p} = (p_0, p_1, p_2, \ldots, p_n) \subseteq (M \times M) - \Delta_M$, there is a corresponding sequence $\overrightarrow{\epsilon} = (\epsilon_0, \epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ where $\epsilon_i \in \{+1, -1\}, i = 1, 2, \ldots, n$, such that for any (x, y) in $(M \times M) - \Delta_M$, it is not the case that both (x, y) and (y, x) are in $\overrightarrow{p}^{\overrightarrow{\epsilon}}$.*

Clearly, not all partial orders can be extended to a total order. The necessary and sufficient condition for a partial order to be extended fully that we provide in Proposition 2.4.1 will be helpful for the study of the Turing degree spectrum of the orderings, which is one research topic that the author would like to pursue in the future.

Chapter 3

Non–Associative Magmas and Their Spaces of Orders

In this chapter, we consider magmas where the operation is not associative. Important examples of such magmas come from knot theory and are called racks and quandles. In these structures, the operation is not associative, but is self-distributive. Racks and quandles were first studied in 1982 by Joyce [25] and Matveev [31]. Some work on fundamental knot quandles and enumeration of finite quandles has been done, however, not much is known about orders on these magmas.

3.1 Definitions of Racks and Quandles

We begin with the definitions, and then we will discuss some important examples of these self-distributive magmas.

Definition 3.1.1. A magma $(Q, *)$ is a *quandle* if the following three axioms are satisfied, where the symbol "$\exists!$" stands for "there is a unique."

(i) $(\forall a)[a * a = a]$, that is, $*$ is idempotent,

(ii) $(\forall b, c)(\exists! a)[a * b = c]$, that is, for every $b \in Q$, the function $*_b : Q \to Q$ defined by $*_b(a) = a * b$ is a bijection, and

(iii) $(\forall a, b, c)[(a * b) * c = (a * c) * (b * c)]$, that is, $*$ is right self-distributive.

Remark: A magma $(Q, *)$ is a *rack* if only the last two axioms are satisfied.

Example 3.1.2. Let G be a group and $g \in G$. For any $a, b \in G$, we define $a*b = a \cdot g$. Then $(G, *)$ forms a rack as it satisfies both rack axioms: for each b, c in G, we can always get $a = c \cdot g^{-1}$ so that $a * b = c$. Moreover, $*$ is right self-distributive as for each a, b, c, both $(a * b) * c$ and $(a * c) * (b * c)$ are equal to $a \cdot g \cdot g$.

If $(G, \cdot)$ is a right orderable group, then $(G, *)$ is right orderable as well. This is because every right order on the group G induces a right order on the rack $(G, *)$. Suppose R is a right order on $(G, \cdot)$. Then if $(a, b) \in R, (a \cdot g, b \cdot g) \in R$.

We define a relation S on $(G, *)$ as follows:

$$(\forall a, b \in (G, *))[(a, b) \in S \Leftrightarrow (a, b) \in R],$$

Then for any $c \in (G, *)$, if $(a, b) \in S$, then $(a*c, b*c) \in S$ since $a*c = a \cdot g, b*c = b \cdot g$, and $(a \cdot g, b \cdot g) \in R$. Therefore, S is a right order on $(G, *)$.

On the other hand, if $g = e_G$ where e_G is the identity element of G, then $(G, *)$ is called a **trivial quandle**. Denote this quandle by $(Q, *)$ and the operation of the equation is simplified to

$$(\forall a, b \in Q)[a * b = a].$$

It is easy to see that for a trivial quandle, every linear ordering of Q is right-invariant with respect to the quandle operation. Moreover, if Q is a countably infinite trivial quandle, we have that $RO(Q)$ is homeomorphic to 2^ω since $RO(Q)$ has no isolated points.

Example 3.1.3. Here we present a rack structure constructed from a free group. The **free rack** $FR(S)$ on a given set S has domain $S \times F(S)$ where $F(S)$ is the free group generated by S and the rack operation is given by

$$(\forall a, b \in S)(\forall w, z \in F(S))[(a, w) * (b, z) = (a, wz^{-1}bz)].$$

Free racks were first introduced in [18].

We first verify the rack axioms for $FR(S)$.

Proposition 3.1.4. *Given a set S, $FR(S)$ satisfies the rack axioms.*

Proof. For any three elements $(a, w), (b, z), (c, y)$ in $FR(S)$, we have:

$$((a, w) * (b, z)) * (c, y) = (a, wz^{-1}bz) * (c, y) = (a, wz^{-1}bzy^{-1}cy),$$

and $((a, w) * (c, y)) * ((b, z) * (c, y)) = (a, wy^{-1}cy) * (b, zy^{-1}cy) = (a, wz^{-1}bzy^{-1}cy)$.

Thus, $((a, w) * (b, z)) * (c, y) = ((a, w) * (c, y)) * ((b, z) * (c, y))$. So, $FR(S)$ satisfies the right self-distributive property.

Additionally, for any two elements $(b, z), (c, y)$ in $FR(S)$, we define $(a, w) = (c, y) * (b, z) = (c, yz^{-1}b^{-1}z)$. Then

$$(a, w) * (b, z) = (c, yz^{-1}b^{-1}z) * (b, z) = (c, yz^{-1}b^{-1}zz^{-1}bz) = (c, y).$$

It is clear that (a, w) is unique.

Since both axioms are satisfied, $FR(S)$ is a rack.

□

Proposition 3.1.5. *Given a totally ordered set S, the free rack $FR(S)$ is right orderable. In fact, each right order on $F(S)$ induces a distinct right order on $FR(S)$.*

Proof. Let P be a right order on $F(S)$. Then for all w, y, and z in $F(S)$,

$$(w, z) \in P \Rightarrow (wy, zy) \in P.$$

We define relation R on $FR(S)$ by requiring that for all (a, w) and (b, z) in $FR(S)$,

$$((a, w), (b, z)) \in R \Leftrightarrow ([(a, b) \in P] \vee [(a = b) \wedge ((e, zw^{-1}) \in P)])$$

where e is the identity of $F(S)$.

Then R is a right order on $FR(S)$ because for any $(c, y) \in FR(S)$, we have $(a, w) * (c, y) = (a, wy^{-1}cy)$ and $(b, z) * (c, y) = (b, zy^{-1}cy)$.

If $((a, w), (b, z)) \in R$ and $a \neq b$, then $(a, b) \in P$, so $((a, wy^{-1}cy), (b, zy^{-1}cy)) \in R$. Thus, $((a, w) * (c, y), (b, z) * (c, y)) \in R$. On the other hand, if $a = b$ and $((a, w), (b, z)) \in R$, then $(e, zw^{-1}) \in P$. We can easily verify that $(e, (zy^{-1}cy) \cdot (wy^{-1}cy)^{-1}) = (e, zw^{-1}) \in P$. Hence, $((a, w) * (c, y), (b, z) * (c, y)) \in R$, and R is a right order on $FR(S)$.

□

Example 3.1.6. For a group $(G, \cdot)$, the **conjugate quandle** of G, in symbol $\mathrm{Conj}(G)$, is $(G, *)$ where the operation $*$ is defined for all elements $a, b \in (G, *)$ by

$$a * b = b^{-1} \cdot a \cdot b.$$

It is routine to verify that $(G,*)$ is a quandle.

We now show that every bi-order $\prec$ on G is a right order on Conj(G), and hence, Conj(G) is orderable.

Let P be a bi-order on G. Then P has the property that

$$(\forall x,c)[(e,x)\in P \Rightarrow (e,c^{-1}xc)\in P)],$$

where e is the identity element of G.

Using P, we define R on Conj(G) as

$$(\forall a,b\in Conj(G))[(a,b)\in R \Leftrightarrow (e,a^{-1}b)\in P)].$$

We claim this order R is right invariant. Let $(a,b)\in R$ and $c\in$ Conj(G). By the definition of $*$, we have:

$$(e,(a*c)^{-1}(b*c))=(e,(c^{-1}a^{-1}c)(c^{-1}bc))=(e,c^{-1}(a^{-1}b)c).$$

As $(a,b)\in R$, $(e,a^{-1}b)\in P$. By the bi-order property of P, it follows that $(e,c^{-1}(a^{-1}b)c)\in P$. Thus, $(e,(a*c)^{-1}(b*c))\in P$, and by the definition of R, $(a*c,b*c)\in R$. Hence, R is a right order on Conj(G).

It was remarked in [11] that in general, not every right order on Conj(G) has to be induced by a bi-order on G. For example, in the case of abelian group G with torsion, G is clearly not bi-orderable; however, Conj(G) is a trivial quandle, so any linear order of the domain is a right order on the quandle. In other words, the quandle admits many right orders while $BiO(G)=\emptyset$ (indeed, in such a group $RO(G)=\emptyset$).

3.2 Orderings on the Free Group

Here, we investigate the space of orders of the conjugate quandle. Specifically, we show that there is a computable free group of infinite rank such that its conjugate quandle does not have a computable right order. As a corollary, we obtain a nice description of the space of right orders of this quandle.

First, let's consider some basic notions. A group G is *free* if there is a set B of elements such that B generates G and there are no nontrivial relations on elements of B. We call B a *basis* for G. All bases for a free group G have the same cardinality, which we call the *rank* of G. We write F_n for the free group of rank n, and F_∞ for the free group of rank $\aleph_0$. The groups F_n, $n \geq 1$, and F_∞ all have computable isomorphic copies. The group F_1 is isomorphic to $(\mathbb{Z}, +)$, so it has two bi-orders. The groups F_n, $n \geq 2$, and F_∞ all have uncountably many bi-orders.

Dabkowska, Dabkowski, Harizanov and Togha proved in [12] that a computable free group G of a finite rank $n > 1$ has a bi-order in every Turing degree. The orders on G are constructed using orders on the quotients of the successive terms of the lower central series of G. Different choices of orders on the quotients of the lower central series yield different bi-orders on G and allow us to encode a set of arbitrary Turing degree into a bi-order on G. Chubb, Dabkowski and Harizanov [8] have recently generalized this result to a large class of computable, finitely presented, residually nilpotent groups. This class includes a variety of important groups such as surface groups, certain nilpotent groups, certain finitely generated one-relator parafree groups, and right-angled Artin groups.

Sikora [39] conjectured that for a free group F_n of finite rank $n > 1$, the space $BiO(F_n)$ is homeomorphic to the Cantor set. It is known that the Cantor set can be embedded into $BiO(F_n)$. McCleary [32] established that for a free group F_n of finite rank $n > 1$, $LO(F_n)$ is homeomorphic to the Cantor set (for another, more recent proof by Navas, see [37]). It remains unknown whether $BiO(F_n)$ for finite $n > 1$ is homeomorphic to the Cantor set.

3.3 Orderings on the Conjugate Quandle of F_∞

A computable free group G of infinite rank with a computable basis has a computable bi-order, and hence its conjugate quandle, which is also computable, has a computable right order. The elements of a basis will generate G, so we can think of them as letters and of the group elements as reduced words on these letters and their inverses. The identity in G cannot be expressed as a nontrivial word on elements of a basis. However, not every computable free group of infinite rank has a computable basis. In fact, it was shown in [6, 33] that every computable isomorphic copy of F_∞ has a basis that is Π_2 in the arithmetic hierarchy, and the result cannot be improved to Δ_2.

Theorem 3.3.1. *There is an isomorphic computable copy G of F_∞ such that its conjugate quandle* $\mathrm{Conj}(G)$ *has no computable right order, and as a result, G has no computable basis.*

Proof. We will build a computable group G isomorphic to F_∞ in stages such that the following requirements are satisfied for all $i \in \mathbb{N}$.

$R_i : \varphi_i^{(2)}$ does not compute a right order on $\mathrm{Conj}(G)$.

The exact strategy for addressing a single requirement R_i is described as follows. At each stage s, we will consider all requirements R_i for $i \leq s$, while determining more and more of the atomic diagram of G. We say R_i *requires attention* if it has not been satisfied. We designate an infinite computable sequence of distinct elements of G:

$$e, a_0, b_0, c_0, \ldots, a_i, b_i, c_i, \ldots,$$

where e is the identity element of G. Having enumerated finitely many sentences in the atomic diagram of G, we continue extending the diagram of G and watching $\varphi_i^{(2)}$ on pairs of elements from $S_i = \{a_i, b_i, c_i\}$. For every $i \in \mathbb{N}$, we will include in G a finitely generated free factor G_i. We start building a computable free group G_i on generators a_i, b_i, c_i and wait until we get to some stage s at which $\varphi_i^{(2)}$ might compute a right order on $\mathrm{Conj}(F_\infty)$, then diagonalize.

For any $g \in G$, and $n \in \mathbb{N}$, we write $g^n = \underbrace{gg\cdots g}_{n}$. For x, y, z, we write $\varphi_{i,s}^{(2)}(x, y) \downarrow = z$ if $i, x, y < s$, and $\varphi_i^{(2)}(x, y)$ halts in fewer than s steps and outputs z.

Construction

Stage 0. Set $G_0 = \{e\}$ and $e * e = e$.

Stage $s+1$: For each $i \leq (s+1)$, continue building free factor G_i whose elements are of length $s+1$. Then there are two cases:

1. Case 1: For each $i \leq (s+1)$, $\varphi_i^{(2)}$ does not need our attention. No further action are needed here. Keep the atomic diagram of G as it is.

2. Case 2: There is some $i \leq (s+1)$ where $\varphi_i^{(2)}$ needs attention. This means that this $\varphi_i^{(2)}$ defined on $S_i \times S_i$ is the characteristic function of a strict linear

ordering $\prec$ of S_i. Without loss of generality, assume, for example, that we have $c_i \prec b_i \prec a_i$; that is, $\varphi_{e,s}^{(2)}(c_i, b_i) = 1$, $\varphi_{e,s}^{(2)}(b_i, c_i) = 0$, $\varphi_{e,s}^{(2)}(c_i, a_i) = 1$, $\varphi_{e,s}^{(2)}(a_i, c_i) = 0$, $\varphi_{e,s}^{(2)}(b_i, a_i) = 1$, $\varphi_{e,s}^{(2)}(a_i, b_i) = 0$, and $\varphi_{e,s}^{(2)}(x, x) = 0$ for $x \in S_i$. At this stage, let $n \in \mathbb{N}$ be some large number so that $(b_i^{-1})^n a_i (b_i)^n$ has not yet been encoded into the atomic diagram of G, and we make $c_i = (b_i^{-1})^n a_i (b_i)^n$. From now on, at later stages, we continue to build the free factor G_i generated just by a_i and b_i. This action will prevent $\varphi_i^{(2)}$ from computing a right order on $\mathrm{Conj}(G)$, and hence, the requirement R_i will be satisfied.

Define G to be the free product of all G_i for all $i \in \mathbb{N}$.

End of the construction.

It is necessary to verify the following lemmas for our construction.

Lemma 3.3.2. *G is computable and it is isomorphic to F_∞.*

Proof. At stages where the requirements $R_i's$ do not need our attention, the atomic diagram of G consists of a finite numbers of statements for each G_i, thus, it is computable. On the other hand, at some stage s where we need to rebuild G_i to stop $\varphi_i^{(2)}$ from computing a right order on $\mathrm{Conj}(G)$, the construction instructs us to choose a sufficiently large n for c_i; then we will not contradict any quantifier-free statements in the atomic diagram of G to which we have already committed. Therefore, the process guarantees that the group G will be computable.

Furthermore, since each G_i is built as a free group with basis (a_i, b_i, c_i) (or (a_i, b_i) in the case R_i requires attention), each G_i is free. Therefore, G, defined to be the free product of the free groups G_i for all $i \in \mathbb{N}$, is isomorphic to F_∞.

□

Lemma 3.3.3. *For all $i \in \mathbb{N}$, R_i is satisfied.*

Proof. Assume otherwise, that is, for some i, R_i is not satisfied and $\varphi_i^{(2)}$ computes a right order $\prec$ on $\text{Conj}(G)$. Then for some stage s, $\varphi_i^{(2)}$ is the characteristic function of the strict linear ordering $\prec$ of S_i and without loss of generality, assume that $c_i \prec b_i \prec a_i$ where $c_i = (b_i^{-1})^n a_i (b_i)^n$ for some large n. Since $b_i \prec a_i$, we have $b_i * b_i \prec a_i * b_i$, so $b_i \prec a_i * b_i$ since $b_i * b_i = b_i$. Hence, $b_i \prec b_i^{-1} a_i b_i$. By continuing multiplying b_i on the right, we get a chain of inequality $b_i \prec b_i^{-1} a_i b_i * b_i$, i.e., $b_i \prec b_i^{-1} b_i^{-1} a_i b_i b_i, \ldots$ As $c_i \prec b_i$, $c_i \prec (b_i^{-1})^n a_i (b_i)^n$ for any n, which gives us the contradiction.

□

□

Corollary 3.3.4. *The space $RO(\text{Conj}(F_\infty))$ is homeomorphic to the Cantor set.*

Proof. Let G be a computable group isomorphic to F_∞ such that the conjugate quandle $\text{Conj}(G)$ has no computable right order. By Theorem 2.3.2, there is a computable binary tree $\mathcal{T}$ and a Turing degree preserving bijection from $RO(\text{Conj}(G))$ to the set of all infinite paths of $\mathcal{T}$. Since $\mathcal{T}$ does not have a computable infinite path, it has no isolated infinite paths. Hence, the space $RO(\text{Conj}(G))$ is homeomorphic to the Cantor set.

□

Chapter 4

Complexity of Natural Algebraic Properties of Magmas

Consider a natural class of algebraic structures that are closed under isomorphisms. An interesting question is to find the exact complexity of determining certain properties within the class; we want to find out how hard it is to determine if a structure exhibits some property from a finitary description of the structure in the form of an algorithm. For instance, in the case of finitely presented groups, the complexity of the word, conjugacy, and isomorphism problems have widely been studied by combinatorial group theorists ([30, 14, 13]).

The complexity of finding algebraic properties of a computable structure can also be represented by the complexity of its *index set*, the set of indices for computable isomorphic copies of the original structure. The study of the index sets in some general class of computable structures is discussed in [5]. There, they describe structures using Scott sentences in infinitary logic. While Scott sentences about the structure provide a bound for the complexity of the index set, the *optimal* Scott sentences give the exact complexity of the index set of the structure.

Before discussing the specific property complexity problems, we revisit the arithmetical hierarchy that we mentioned in the introduction and introduce the notion of *completeness*; we will go over some well-known sets with known results about their completeness within the arithmetical hierarchy.

4.1 Basic Notions

One way to access the precise complexity of a set is to determine its location in the *arithmetical hierarchy* of sets. As we discussed in the introduction, formula complexity only gives an upper bound; however, there are more refined tools that provide precise location of the complexity on the hierarchy. We describe a few such methods below.

Definition 4.1.1. Let A, B be sets. A is defined to be *many-one reducible (m-reducible)* to B, denoted by $A \leq_m B$, if there is a computable function f such that $f(A) \subseteq B$ and $f(\overline{A}) \subseteq \overline{B}$, i.e., $x \in A$ if and only if $f(x) \in B$.

We can now give a formal definition for *completeness*, an essential notion used to understand the precise complexity of a given set:

Definition 4.1.2. Let A be a set. We say A is Σ^0_n-*complete* (or Π^0_n-*complete*) if

i. $A \in \Sigma^0_n$ (Π_n), and

ii. A is Σ^0_n *m-hard* (or Π^0_n *m-hard*), i.e., $B \leq_m A$ for every $B \in \Sigma^0_n$ (Π^0_n).

For example, the halting set, $K = \{e : e \in dom(\varphi_e)\}$, is a well-known Σ^0_1-complete set. Other well-known examples that will be useful for us later on in this chapter are $FIN = \{x : W_x \text{ is finite}\}$, which is Σ^0_2-complete, and $INF = \omega - FIN = \{x : W_x \text{ is infinite}\}$, which is Π^0_2-complete.

4.2 Commutative Magmas

We begin by studying the complexity of detecting commutativity in magmas. Recall that a magma $(M, \cdot)$ is *commutative*, or *Abelian*, if for any a, b in M, $a \cdot b = b \cdot a$.

Proposition 4.2.1. *Identifying an Abelian magma is* Π^0_1*-complete within the class of computable magmas.*

Proof. The property of being an Abelian magma $(M, \cdot)$ is characterized by

$$(\forall a, b \in M)[a \cdot b = b \cdot a],$$

which is a Π^0_1 formula since equality is computable in the class of computable magmas.

We noted above that K is Σ^0_1-*complete*, so we know that its complement, $\overline{K} = \{e \mid \varphi_e(e) \uparrow\}$, is Π^0_1-complete. Thus, to show the completeness in our proposition, we m-reduce $\overline{K}$ to the index set of Abelian magmas.

For each $e \in \omega$, we will construct the atomic diagram for a magma M_e such that M_e is isomorphic to an Abelian magma if $e \in \overline{K}$ and is isomorphic to a non-Abelian magma otherwise.

Without loss of generality, we can assume that the domain of M_e is the set of all natural numbers. We define a coding map $(\cdot) : \mathbb{N} \to \mathbb{N}$ and generate the atomic diagram for M_e in stages as followings:

Construction:

Stage 0: Add 0 to the domain. Set $0 \cdot 0 = 0$ and add $(0, 0, 0)$ to the diagram of $M_{e,0}$.

Stage s+1: Let the set $M_{e,s} = \{0, m_1, m_2, \ldots, m_{k_s}\}$ of natural numbers indexed by natural numbers represent the sth approximation of M_e at the end of stage s.

Let j be the least index such that m_j has not been coded into M_e. Add m_j to the domain of the atomic diagram.

For all m_i's already enumerated into M_e, consider the product of any two elements and let k, l be the least indices of elements in M_e such that at least one of $m_k \cdot m_l$ or $m_l \cdot m_k$ has not been assigned a value. From here, we take different actions depending on the value of $\varphi_{e,(s+1)}(e)$.

<u>Case 1</u>: $\varphi_{e,(s+1)}(e) \uparrow$. If values of both products, $m_k \cdot m_l$ and $m_l \cdot m_k$, have not decided, set $m_k \cdot m_l = m_l \cdot m_k$ and assign m_p, where m_p is the least element that has not been enumerated into M_e, to be the product. If either $m_k \cdot m_l$ or $m_l \cdot m_k$ has been encoded using some m_p, set the other product to the same m_p.

<u>Case 2</u>: $\varphi_{e,(s+1)}(e) \downarrow$. If values of both products, $m_k \cdot m_l$ and $m_l \cdot m_k$, have not decided, use distinct m_j and m_n ($m_j \neq m_n$) not yet in the domain and set $m_k \cdot m_l = m_j$ and $m_l \cdot m_k = m_n$. If either $m_k \cdot m_l$ or $m_l \cdot m_k$ has been encoded using some m_p, set the other product to some m_q not yet enumerated into M_e.

For the rest of the atomic diagram of M_e, for any m_i, m_j that are already enumerated into M_e, assign values to $m_i \cdot m_j$ if needed.

End of the construction.

Clearly $M_e = \cup_{s \in \omega} M_{e,s}$ is a computable magma and M_e is Abelian if and only if

$e \in \overline{K}$.

We conclude that within the class of computable magmas, detecting commutativity is Π^0_1-complete.

□

4.3 Markov Properties

The result for Abelian magmas in the previous theorem makes us wonder what other natural properties would have the same level of complexity in identifying them in computable magmas. It turns out being Abelian belongs to a large class of properties known as **Markov properties** [1].

Definition 4.3.1. Let $\mathcal{C}$ be a class of magmas. An abstract property P is *Markov* with respect to $\mathcal{C}$ if there is a magma $M_+ \in \mathcal{C}$ that satisfies the property, and there is a magma M_- where for any magma N, if there is an injective homomorphism from M_- into N, N fails to have property P.

In this section, we show that Markov properties are not *recursively recognizable* ([1]) in the class of computable magmas. In other words, there is no effective algorithm to decide from the atomic diagram of a computable magma whether the magma exhibits a certain Markov property or not.

Adian and Rabin showed that Markov properties are not recursively recognizable within the class of finitely presented groups [1]. Bilanovic, Chubb, and Roven proved that detecting Markov properties is Π^0_2-hard in the class of recursively presented groups, and it is Π^0_1-hard in the class of computable groups [3]. Here, we investigate the similar question in a more general type of algebraic structure, the class of computable magmas.

Theorem 4.3.2. *Let P be a Markov property for computable magmas. Then if M_+ can be chosen to be infinite, detecting P in this class of magmas is Π^0_1-hard.*

Proof. We take $\mathbb{N}$ as the domains of M_+ and M_-. We can assume here that M_- is also infinite because M_- can be embedded into the direct product of itself with the computable additive group of integers, $M_- \times \mathbb{Z}$. We denote id here as an "identity" element for M_- in the sense that id will act as an identity but it is not necessarily an element of M_- (In the case where M_- has an identity element, we will use this identity and denote it by id).

We use the computable atomic diagrams of $(M_+, \cdot)$ and $(M_-, *)$ in a construction by finite approximation of the computable atomic diagram of a magma $(M_e, \star)$ with domain $\mathbb{N}$ so that M_e is isomorphic to M_+ when $\varphi_e(e)\uparrow$ and isomorphic to $M_+ \times M'_-$, where $M'_- = (M_- \cup \{id\})$, if $\varphi_e(e)\downarrow$. In other words, we ensure that M_e satisfies P if and only if $e \in \overline{K}$. Because we already know that $\overline{K}$ is a Π^0_1-complete set, we will conclude that the detection of P in the class of computable magmas is Π^0_1-hard.

Let $M_+ = \{m_0, m_1, m_2, \ldots\}$ and $M_- = \{n_0, n_1, n_2, \ldots\}$ be computable enumerations of all distinct elements of two magmas witnessing that P is a Markov property for the class of computable magmas.

Construction:

Stage 0: Use 0 to code (m_0, id), i.e., set $\langle (m_0, id) \rangle = 0$. As $(m_0, id) \star (m_0, id) =$

$(m_0 \cdot m_0, id)$, set

$$\langle (m_0 \cdot m_0, id) \rangle = \begin{cases} 1 & \text{if } m_0 \cdot m_0 \neq_+ m_0, \\ 0 & \text{if } m_0 \cdot m_0 =_+ m_0. \end{cases}$$

Then add $(0, 0, 1)$ or $(0, 0, 0)$ to the atomic diagram of M_e. (Equality in M_+ is denoted by $=_+$.)

Stage s+1: Since we code differently based on the existence of value of $\varphi_e(e)$, we have three cases:

1. Case 1: $\varphi_{e,(s+1)}(e) \uparrow$, in other words, $e \notin K_{s+1}$. Let ℓ be the least index of an element of M_+ where (m_ℓ, id) has not been assigned a code yet, and set (m_ℓ, id) to be the least number we have not yet used as a code. Now, let j be the least index of an element of M_+ such that there exists a $k \leq j$ where the product $(m_j, id) \star (m_k, id)$ has not been defined in the atomic diagram of M_e. For each such $k \leq j$, if we already have a code for $(m_j \cdot m_k, id)$, add the tuple $(\langle (m_j, id) \rangle, \langle (m_k, id) \rangle, \langle (m_j \cdot m_k, id) \rangle)$ to the diagram; otherwise, we assign the next available number, say p, to $(m_j \cdot m_k, id)$ and then we add the tuple $(\langle (m_k, id) \rangle, \langle (m_j \cdot m_k, id) \rangle, p)$ to the diagram.

2. Case 2: $\varphi_{e,(s+1)}(e) \downarrow$ and $\varphi_{e,s}(e) \uparrow$. So e has just entered the halting set at stage $(s + 1)$. Before this stage, we have been encoding elements of M_+ into M_e. When $\varphi_e(e) \downarrow$, we turn our focus to the second component of $M_+ \times M_- = M_e$.

   For all p, whenever we have already enumerated (m_p, id), we start coding (m_p, n_0) with fresh natural number codes. To guarantee the closure the operation $(\star)$ on magma M_e, for each $\langle (m_j, id) \rangle$ and $\langle (m_k, n_0) \rangle$ that are already coded, we assign new numbers to all new tuples $(\langle (m_j, id)\rangle, \langle (m_k, n_0)\rangle, \langle (m_j \cdot m_k, n_0)\rangle)$ whenever needed.

3. <u>Case 3</u>: $\varphi_{e,(s+1)}(e)\downarrow$ and $\varphi_{e,s}(e)\downarrow$, so e appeared in the halting set at a previous stage. Let k, j be the least indices for which (m_k, id) and (m_0, n_j) have not been enumerated into M_e and assign new numbers to them. For any x, y in M_+ and u, v in $(M_- \cup \{id\})$ that have been enumerated in M_e at this stage, code $\langle (x \cdot u, y * v)\rangle$ if needed and add the coded tuple $(\langle (x, y)\rangle, \langle (u, v)\rangle, \langle (x \cdot u, y * v)\rangle)$ to the atomic diagram of M_e.

End of the construction.

It is clear from the construction that M_e is a computable magma, and it is isomorphic to $M_+ \times \{id\} \cong M_+$, (and so M_e has property P), if $e \in \overline{K}$; otherwise, it is isomorphic to $M_+ \times M'_-$, which fails to have property P since M_- embeds into M_e via M'_-. Thus, identifying P in computable magmas is Π^0_1-hard.

□

Since we showed that the index set of a computable magma that satisfies a Markov property is Π^0_1-hard, if the Markov property can be characterized by a Π^0_1 formula, we obtain the completeness for the complexity of the index set. For example, the following corollary summaries the precise complexity of a few Markov properties.

Corollary 4.3.3. *Within the class of computable magmas, the index set of magmas that satisfy any of the following properties is* Π^0_1*-complete:*

(i) Idempotence

(ii) Right self-distributivity

(iii) Associativity

(iv) Commutativity

(v) Orderability.

Proof. (i) The condition for a magma $(M, \cdot)$ to be idempotent is expressed by the following Π^0_1 formula:

$$(\forall a \in M)[a \cdot a = a].$$

Take M_+ to be some infinite idempotent magma ($\mathrm{Conj}(F_\infty)$ for instance) and M_- to be some non-idempotent magma. Then for any magma N, such that M_- can be embedded into N, N will fail to be idempotent. Thus, being idempotent is a Markov property. Applying Theorem 4.3.2, we conclude that detecting idempotence in computable magmas is Π^0_1-complete.

(ii) Right self-distributivity in a magma $(M, \cdot)$ is characterized by

$$(\forall a, b, c \in M)[(a \cdot b) \cdot c = (a \cdot c) \cdot (b \cdot c)],$$

which is a Π^0_1 formula.

Choose M_+ to be an infinite right self-distributive magma, for example, the free rack $F(S)$, and M_- to be some magma that is not right self-distributive (any non trivial group would work). Using similar argument and the same theorem, we see that right self-distributivity is also a Markov property, and thus, the index set for right self-distributive magmas is Π^0_1-complete within the class of computable magmas.

(iii) A magma $(M, \cdot)$ is associative if

$$(\forall a, b, c \in M)[(a \cdot b) \cdot c = a \cdot (b \cdot c)].$$

As this is a Π_1^0 formula, take M_+ to be $(\mathbb{Z}, +)$ and M_- to be $\mathrm{Conj}(F_\infty)$; applying Theorem 4.3.2, we reach the same conclusion for the index set of associative magmas.

(v) Based on Corollary 2.4.3, a magma $(M, \cdot)$ is orderable if

$$(\forall \overrightarrow{p} \subseteq M^\star)(\exists \overrightarrow{\epsilon} \in \{\pm 1\}^{|\overrightarrow{p}|})(\forall (x,y) \in M^\star)[(x,y) \wedge (y,x)) \notin \overrightarrow{p}^{\overrightarrow{\epsilon}}].$$

where $M^\star$ denotes $(M \times M) - \Delta_M)$.

Since the existential quantifier here is bounded, it will not affect the complexity level of the formula and can be ignored. Furthermore, the computability of the magma allows us to enumerate and check the elements in $\overrightarrow{p}^{\overrightarrow{\epsilon}}$ computably; thus, the formula describing the orderability characterization of a computable magma is in fact a Π_1^0 formula.

Take M_+ to be an infinite orderable magma, a trivial quandle based on $\mathbb{Z}$ for example, and M_- to be the crystallographic group Γ, the non-orderable group that we discussed at the beginning of Chapter 2. Applying Theorem 4.3.2, we conclude that detecting orderability in computable magmas is Π_1^0-complete.

□

A self distributive magma is also known as a **shelf** and it has applications in the study of knot theory. Hence, the previous corollary leads us to the following corollary.

Corollary 4.3.4. *Identifying a shelf within the class of computable magmas is* Π_1^0-*complete.*

4.4 Properties with Higher Complexity Levels

In this section, we discuss the detection problem of properties that are m-complete at higher levels on the arithmetical hierarchy than those in the previous section. We will consider left-inverse, unique left-inverse, and identity properties.

4.4.1 Left Inverse Property

As magmas are not required to have an identity, the definition of *left inverese* element needs to be modified to accommodate a more general setting. We define the left inverse property as follows.

Definition 4.4.1. A magma $(M, \cdot)$ has the *left inverse* property if for all b, c in M, there exists an element a in M such that $a \cdot b = c$.

The left-inverse property is a Markov property. Our general theorem about Markov properties in the previous section shows that they are Π^0_1-hard, but this is just a lower bound. The following proposition provides the exact complexity level of the left-inverse property.

Proposition 4.4.2. *The index set of magmas with the left-inverse property* $(M, \cdot)$ *is* Π^0_2*-complete in the class of computable magmas.*

Proof. The property of having left-inverse for a magma $(M, \cdot)$ is characterized by

$$(\forall b, c \in M)(\exists a \in M)[a \cdot b = c],$$

which is a Π^0_2 formula since equality is computable in the class of computable magmas.

To show completeness, we reduce the Π^0_2-complete set INF to the index set of left-inverse magmas.

For each $e \in \omega$, we construct the atomic diagram for a magma M_e in such a way that M_e satisfies the left-inverse property if $e \in INF$ (i.e., W_e is infinite) and fails to have the property if $e \notin INF$. To guarantee that M_e has the left-inverse property, we need to make sure that all elements in the domain of M_e appear in each column of its multiplication table. On the other hand, to stop M_e from satisfying the left-inverse property, we need to make sure that at least one element is missed in some column. We will be able to achieve this by repeating a cycle of a finite number of elements of M_e in some column. Since M_e is infinite and the action is taken when W_e is finite, not all elements of M_e will show up in this column. Thus, the missing elements in the column are the ones that ensure that M_e fails to have the left-inverse property.

Take $\mathbb{N}$ to be the domain of M_e. We build the multiplication table of $(M_e, \cdot)$ in stages as described below. At the end of each stage, as usual, we get a finite set $M_{e,s} = \{0, m_1, m_2, \ldots, m_{k_s}\}$ of natural numbers as the sth approximation of M_e, along with a finite approximation of the atomic diagram of M_e, i.e., its multiplication table.

Construction:

Stage 0: Add 0 to the domain. Set $0 \cdot 0 = 0$.

Stage s+1: Let m_j be the least natural number not yet in the domain of M_e. We will create a new row and column for m_j.

Let i be the index of an element m_i of M_e with the shortest column.

<u>Case 1</u>: $W_{e,(s+1)} - W_{e,s} \neq \emptyset$.

Add the next available number m_k (i.e., it has not shown up in the column) to the next row of the column. Then the corresponding element that represents that row, let's say m_j, is the left inverse of the pair (m_i, m_k). Thus, we add (m_j, m_i, m_k) to the atomic diagram of M_e.

Case 2: $W_{e,(s+1)} - W_{e,s} = \emptyset$.

We do not add new element from the domain of M_e to this column; instead, we repeat the first entry of the column in the next row of the column.

End of the construction.

Based on the construction, it is clear that $M_e = \cup_{s\in\omega} M_{e,s}$ is a computable magma. Furthermore, when W_e is infinite, it is guaranteed that there is a left inverse for any pair of elements in M_e, and if W_e stops getting new elements eventually (i.e., if W_e is finite), M_e will fail to satisfy the left-inverse property. Thus, INF is reduced to the Π^0_2 index set of magmas with left inverse property. In other words, identifying left inverse property is Π^0_2-complete in the class of computable magmas.

□

4.4.2 Unique Left Inverse Property

In algebra, we often encounter structures with the unique left inverse property. Interestingly, we find out that the addition of "uniqueness" to the left inverse property does not raise the level of complexity of the index set of the structure.

Definition 4.4.3. We say a magma $(M, \cdot)$ satisfies *unique left inverse* property if for all b, c in M, there is a unique element a in M such that $a \cdot b = c$.

Remark: A right self-distributive magma that satisfies the unique left inverse property is referred to as a **rack** in the literature. And an idempotent rack is called a **quandle**, the space of orderings of which is discussed in a previous chapter.

Proposition 4.4.4. *The index set of magmas* $(M, \cdot)$ *having unique left-inverse is* Π^0_2*-complete in the class of computable magmas.*

Proof. We take a different approach to prove this proposition. We will show that the problem of identifying the left-inverse property, denoted (LI), is *m-equivalent* to the problem of identifying the unique left-inverse property, denoted $(!LI)$.

Recall that we say A is a *many-one reducible (m-reducible)* to B (denoted $A \leq_m B$) if there is a computable function f such that $f(A) \subseteq B$ and $f(\overline{A}) \subseteq \overline{B}$, i.e., $x \in A$ if and only if $f(x) \in B$. First, we show that $(!LI) \leq_m (LI)$ via a computable function $h : \mathbb{N} \to \mathbb{N}$ so that $M_e \models (!LI) \Leftrightarrow M_{h(e)} \models (LI)$.

Let $m_{i,j}$ denote the element of M_e in the ith row and jth of the multiplication table. Note that $(!LI)$ is equivalent to having each element of M_e appearing *exactly once* in each column of the multiplication table and that (LI) is equivalent to having each element of M_e showing up *at least once* in each column of the table.

Our strategy is the following: We will build $N = M_{h(e)}$ so that in each column of the multiplication table of N, each entry in each row is the same as the corresponding entry in the multiplication table of M_e, unless there are at least two rows with the same value in the corresponding column in the multiplication table of M_e. When we observe a repetition of entries in a column in the table of M_e, the corresponding entries in the table of N are filled with a value of some preceding entry in the same column and we make sure that the value of the repeated entries will not show up at all in the corresponding column of N.

Using this strategy, we build the multiplication table for $N = M_{h(e)}$ in stages below:

Construction:

Stage 0: Set $n_0 = m_0$ and add $(n_0, n_0, m_{0,0})$ to the diagram of N.

Stage s: Set $n_s = m_s$. Following the strategy we describe above, we fill all entries in the $(s+1)$st row and $(s+1)$st column in the table of N.

There are two possible scenarios:

1. $M_{e,s}$ has some $i, j, k \leq s$ so that $m_i \cdot m_k = m_j \cdot m_k$ but $m_i \neq m_j$ and i is the least index where we observe the repetition of values on the kth column. Then for any of these i, j, $n_{i,k} = n_{j,k} = n_{i-1,k}$. In the case where i happens to be 0, for all j such that $m_{0,k} = m_{j,k}$, we assign some natural number that is different from $m_{0,k}$ to $n_{0,k}$ and all $n_{j,k}$'s.

   Moreover, for later stages, if the value of $m_{i,k}$ shows up again at some succeeding rows in the same k th column in the table of M, we make sure not to use this value but assign the value of $n_{i-1,k}$ to the corresponding entry in the table of N.

2. Otherwise, for $i, j \leq s, n_{i,j} = m_{i,j}$.

End of the construction.

Next, we prove that $(LI) \leq_m (!LI)$ by giving a computable function $f : \mathbb{N} \to \mathbb{N}$ so that $M_e \models (LI) \Leftrightarrow M_{f(e)} \models (!LI)$. We use even numbers to represent elements of M_e in $G = M_{f(e)}$ aiming to mimic the behavior of M_e in G to some extent. Our overall strategy is that we will introduce auxiliary elements into the multiplication table of $M_{f(e)}$; they consist of odd numbers which are "woven" into the even numbers representing elements of M_e. The purpose of including the new elements is to prevent the duplication of elements of M_e in each column in the case where $M_e \models (LI)$, but $M_e \not\models (!LI)$.

The domain of $G = M_{f(e)}$ is all natural numbers. Let a sequence of even numbers $\{m_0, m_1, m_2, \ldots\}$ be the representation of distinct elements of M_e. We build the multiplication table for G in stages as following:

Construction:

Stage 0: Set $g_0 = m_0$ and $g_1 = 1$. Add $(g_0, g_0, m_{0,0})$ and $(g_0, g_1, m_{0,1})$ to the diagram. If $m_{1,0} \neq m_{0,0}$, set $g_1 \cdot g_0 = m_{1,0}$ and add $m_{1,0}$ to the table; otherwise, $g_{1,0} = g_1$ and add g_1 instead. For the second column (the column of the first odd element g_1), to keep things simple, we treat g_1 as the right identity element, i.e., we fill up each entry in this column with the corresponding value from the 0^{th} column—the furthest left one in the table.

We illustrate our construction of the first stage in the tables in Figure 4.1, which captures both possible scenarios for the multiplication table of G at the end of stage 0.

Table 4.1: $m_{0,0} \neq m_{1,0}$

	$g_0 = m_0$	$g_1 = 1$
$g_0 = m_0$	$m_{0,0}$	m_0
$g_1 = 1$	$m_{1,0}$	1

Table 4.2: $m_{0,0} = m_{1,0}$

	$g_0 = m_0$	$g_1 = 1$
$g_0 = m_0$	$m_{0,0}$	m_0
$g_1 = 1$	1	1

Figure 4.1: Both scenarios for the multiplication table of G_0

Stage s: Set $g_{2s} = m_s$ and $g_{2s+1} = 2s + 1$. We need to fill entries on the new $(2s + 1)$st and $(2s + 2)$nd rows and $(2s + 1)$st and $(2s + 2)$nd columns of the table of G using the strategy outlined above. For each odd column $(2k + 1)$st $(k \leq s)$ in the multiplication table of G, we use the next element from the $(k + 1)$st column on the multiplication table of M_e, let's say $m_{i,k+1}$ to fill the next open entry. However, if the value of $m_{i,k+1}$ already shows up in the $(2k + 1)$st column of the table of G from previous stages, we use the next odd number that has not been enumerated into the $(2k + 1)$st column to fill that entry. On the other hand, for each even column $(2k + 2)$nd $(k \leq s)$ in the table of G, set $m_i \cdot (2k + 1) = m_i$ for all $i \leq (2s + 1)$ and add the corresponding product to the table.

The tables below in Figure 4.2 show some possibilities of what the multiplication table of G would look like at the end of stage 2 after the strategy is carried out:

Table 4.3: $m_{2,0} \neq m_{0,0}$ and $m_{3,0} \neq m_{0,0}$

	m_0	1	m_1	3
m_0	$m_{0,0}$	m_0	$m_{0,1}$	m_0
1	1	1	$m_{1,1}$	1
m_1	$m_{2,0}$	m_1	$m_{2,1}$	m_1
3	$m_{3,0}$	3	$m_{3,1}$	3

Table 4.4: $m_{2,0} = m_{0,0}$ and $m_{3,0} \neq m_{0,0}$

	m_0	1	m_1	3
m_0	$m_{0,0}$	m_0	$m_{0,1}$	m_0
1	1	1	$m_{1,1}$	1
m_1	3	m_1	$m_{2,1}$	m_1
3	$m_{3,0}$	3	$m_{3,1}$	3

Table 4.5: $m_{2,0} \neq m_{0,0}$ and $m_{3,0} = m_{0,0}$

	m_0	1	m_1	3
m_0	$m_{0,0}$	m_0	$m_{0,1}$	m_0
1	1	1	$m_{1,1}$	1
m_1	$m_{2,0}$	m_1	$m_{2,1}$	m_1
3	3	3	$m_{3,1}$	3

Table 4.6: $m_{2,0} = m_{3,0} = m_{0,0}$

	m_0	1	m_1	3
m_0	$m_{0,0}$	m_0	$m_{0,1}$	m_0
1	1	1	$m_{1,1}$	1
m_1	3	m_1	$m_{2,1}$	m_1
3	5	3	$m_{3,1}$	3

Figure 4.2: Possible multiplication tables of G_2

Notice that in the four tables above, as we want to focus our action on the first column only, we show examples where elements are all different in the last three columns. But of course, we would apply the same strategy if there were repetitions in the corresponding columns in the multiplication table M_e.

End of the construction.

We complete our proof by verifying the following lemmas:

Lemma 4.4.5. *$(!LI) \leq_m (LI)$ via the computable function h.*

Proof. Suppose M_e satisfies the unique left-inverse property. Then each column of the multiplication table of M_e contains all elements of M_e exactly once. Then by our construction, $M_{h(e)} = M_e$ and thus, in the multiplication table of $M_h(e)$, all elements show up in each column, and $M_{h(e)}$ satisfies the left-inverse property.

On the other hand, if M_e does not satisfy the unique left-inverse property, that implies either M_e misses at least one of its element in some column on its multiplication table or some element shows up at least more than once in some column on the table. If it is the former case, our strategy yields $M_{h(e)} = M_e$ as the same missing element will not appear in the corresponding column on the multiplication table of $M_{h(e)}$ and $M_{h(e)}$ will fail to have the left-inverse property. In the latter case, where some element $m_{i,j}$ shows up more than once in some jth column of M_e, the construction for $M_{h(e)}$ will assign a different number to all entries with value of $m_{i,j}$. That means the jth column of $M_{h(e)}$ does not contain the element $m_{i,j}$, and therefore, $M_{h(e)}$ does not satisfy the left-inverse property.

As $M_e \models (!LI) \Rightarrow M_{h(e)} \models (LI)$ and $M_e \not\models (!LI) \Rightarrow M_{h(e)} \not\models (LI)$, h witnesses $(!LI) \leq_m (LI)$.

□

Lemma 4.4.6. *$(LI) \leq_m (!LI)$ via the computable function f.*

Proof. Suppose M_e satisfies the left-inverse property. Then all elements of M_e appear in each column of its multiplication table. Moreover, our construction of $M_{f(e)}$ guarantees that in each column of $M_{f(e)}$, we use every element in the corresponding column of M_e and each of these elements and each of the odd numbers show up exactly once. Hence, each column of $M_{f(e)}$ consists of all elements of the magma (all natural numbers) without any repetition. In other words, $M_{f(e)}$ satisfies the unique left-inverse property.

On the other hand, suppose M_e does not satisfy the left-inverse property. Then there exists at least one column in the multiplication table of M_e that does not include at least one element of M_e, let's say m_i is missing on the kth column. Since $M_{f(e)}$ only gets its even elements from M_e, m_i will also be missed on the corresponding kth column in the multiplication table of $M_{f(e)}$. Thus, $M_{f(e)}$ does not satisfy the left-inverse property and as a result, fail the unique left-inverse property as well.

Since $M_e \models (LI) \Rightarrow M_{f(e)} \models (!LI)$ and $M_e \not\models (LI) \Rightarrow M_{f(e)} \not\models (!LI)$, f witnesses $(LI) \leq_m (!LI)$.

□

We claim $(!LI) \leq_m (LI)$ via h, and $(LI) \leq_m (!LI)$ via f. Thus, $(LI) \equiv_m (!LI)$, and since (LI) is Π^0_2-complete, $(!LI)$ is Π^0_2-complete as well.

□

Remark: The property of having unique left-inverse for a magma $(M, \cdot)$ can be expressed most naturally by the following formula:

$$(\forall b, c, d \in M)(\exists a \in M)[(a \cdot b = c) \wedge [(d \cdot b = c) \Rightarrow d = a]],$$

which is a Π_2^0 statement. Hence, we can also prove the completeness of the complexity of detecting the unique left-inverse property within the class of computable magmas by using the conventional approach we discuss at the beginning of this chapter: we can reduce the Π_2^0-complete set INF to the index set of computable magmas that satisfy the unique left inverse property.

4.4.3 Identity Property

The last property we investigate in this chapter is the *identity* property.

Definition 4.4.7. An element id of a magma $(M, \cdot)$ serves as the *identity* for the magma if for all a in M, $a \cdot id = id \cdot a = a$.

The existence of an identity element leads to the notion of inverse elements, which is one of the key features of groups. Nevertheless, there are algebraic structures with no identity such as semi-groups and the cross product of vectors in $\mathbb{R}^3$. In this subsection, we study the complexity of determining whether a computable magma has an identity or not.

Proposition 4.4.8. *The index set of magmas with an identity element* $(M, \cdot)$ *is* Σ_2^0-*complete in the class of computable magmas.*

Proof. Magma $(M, \cdot)$ has an identity if and only if

$$(\exists\, id \in M)(\forall a \in M)[a \cdot id = id \cdot a = a],$$

which is a Σ_2^0 formula.

For completeness, we reduce the Σ_2^0-complete set FIN to the index set of magmas with identity elements.

For each $e \in \omega$, we build the multiplication table for a magma M_e so that M_e has an identity element if $e \in FIN$ (i.e., if W_e is finite) and fails to have the property if $e \notin FIN$.

The domain of M_e will be all $\mathbb{N}$. We build the multiplication table of $(M_e, \cdot)$ in stages as described below. At the end of each stage, we get a set $M_{e,s} = \{0, m_1, m_2, \ldots, m_s\}$ of natural numbers as the sth approximation of M_e. The general strategy is that $M_{e,s}$ will look like it has an identity if there is no new element entered W_e at stage s; otherwise, $M_{e,s}$ will look like it does not have an identity at this stage.

Construction:

Stage 0: Pick 0 to be our identity element at this stage and add 0 to the domain of M_e. Then the first row and first column of the table is reserved for products of the first identity candidate, element 0, and all other elements of M_e (unless 0 stops being an identity at some later stage). As 0 is supposed to behave like the identity element (for now, at least), set $0 \cdot 0 = 0$ and add $(0, 0, 0)$ to the multiplication table.

Stage s+1: Add m_{s+1} to the domain of M_e. Add a new row and a new column for m_{s+1}. The result of products between m_{s+1} and all elements of $M_{e,(s+1)}$ depends on the size of W_e at this stage and the previous stage.

There are four possible scenarios:

1. <u>Case 1</u>: $W_{e,(s+1)} - W_{e,s} \neq \emptyset$ and $W_{e,s} - W_{e,(s-1)} = \emptyset$.

   Suppose that some m_k $(k \leq s)$ is our current candidate for an identity of $M_{e,s}$. Set $m_{s+1} \cdot m_k = m_{s+1}$ and add the tuple to the atomic diagram. Assign some other natural number that is different from m_{s+1}, let's say m_s, to the product

$m_k \cdot m_{s+1}$ and add that to the table. This action stops m_k from being the identity of M_e and at this stage, $M_{e,(s+1)}$ no longer has an identity element. We use numbers from $\{0, m_1, m_2, \ldots, m_{s+1}\}$ to fill up other entries in $(s+1)$ st row and $(s+1)$ st column.

2. Case 2: $W_{e,(s+1)} - W_{e,s} \neq \emptyset$ and $W_{e,s} - W_{e,(s-1)} \neq \emptyset$.

   Inheriting from Case 1, $W_{e,s}$ does not have an identity. We need to make sure that the new element m_{s+1} will not become the identity. We set $0 \cdot m_{s+1} = m_{s+1}$ and $m_{s+1} \cdot 0 = m_s$. Fill the rest of the table with natural numbers from $\{0, m_1, m_2, \ldots, m_{s+1}\}$.

3. Case 3: $W_{e,(s+1)} - W_{e,s} = \emptyset$ and $W_{e,s} - W_{e,(s-1)} \neq \emptyset$.

   At the end of stage s, $W_{e,s}$ did not have an identity. We assign values to entries in $(s+1)$ st row and $(s+1)$ st column of the table of $W_{e,(s+1)}$ so that m_{s+1} is a new candidate for the identity of M_e. Set $0 \cdot m_{s+1} = m_{s+1} \cdot 0 = 0$, and for all $0 \leq j \leq (s+1)$, set $m_j \cdot m_{s+1} = m_{s+1} \cdot m_j = m_j$.

4. Case 4: $W_{e,(s+1)} - W_{e,s} = \emptyset$ and $W_{e,s} - W_{e,(s-1)} = \emptyset$.

   Suppose that some m_k $(k \leq s)$ is the "identity" of $M_{e,s}$. We want to preserve m_k as the possible identity at stage $s+1$. Set $m_k \cdot m_{s+1} = m_{s+1} \cdot m_k = m_{s+1}$ and add the corresponding tuples to the diagram. Then we complete the table of $M_{e,(s+1)}$ with values from $\{0, m_1, m_2, \ldots, m_{s+1}\}$ as usual.

The multiplication tables of $W_{e,(s+1)}$ for all four cases can be illustrated as follows:

TABLE 4.7: $W_{e,(s+1)} - W_e \neq \emptyset$ and $W_{e,s} - W_{e,(s-1)} = \emptyset$

	0	...	$\mathbf{m_k}$	...	$\mathbf{m_{s+1}}$
0	0	...	0	...	m_0
⋮	⋮	⋮	⋮	⋮	⋮
$\mathbf{m_k}$	0	...	m_k	...	$\mathbf{m_s}$
⋮	⋮	⋮	⋮	⋮	⋮
$\mathbf{m_{s+1}}$	m_1	...	$\mathbf{m_{s+1}}$	...	m_2

TABLE 4.8: $W_{e,(s+1)} - W_{e,s} \neq \emptyset$ and $W_{e,s} - W_{e,(s-1)} \neq \emptyset$

	0	...	$\mathbf{m_{s+1}}$
0	0	...	$\mathbf{m_{s+1}}$
⋮	⋮	⋮	⋮
$\mathbf{m_{s+1}}$	$\mathbf{m_s}$	...	m_0

TABLE 4.9: $W_{e,(s+1)} - W_e = \emptyset$ and $W_{e,s} - W_{e,(s-1)} \neq \emptyset$

	0	...	m_j	...	$\mathbf{m_{s+1}}$
0	0	...	m_0	...	0
⋮	⋮	⋮	⋮	⋮	⋮
m_j	0	...	m_1	...	$\mathbf{m_j}$
⋮	⋮	⋮	⋮	⋮	⋮
$\mathbf{m_{s+1}}$	0	...	$\mathbf{m_j}$	...	m_{s+1}

TABLE 4.10: $W_{e,(s+1)} - W_{e,s} = \emptyset$ and $W_{e,s} - W_{e,(s-1)} = \emptyset$

	0	...	$\mathbf{m_k}$	...	$\mathbf{m_{s+1}}$
0	0	...	0	...	m_0
⋮	⋮	⋮	⋮	⋮	⋮
$\mathbf{m_k}$	0	...	m_k	...	$\mathbf{m_{s+1}}$
⋮	⋮	⋮	⋮	⋮	⋮
$\mathbf{m_{s+1}}$	m_1	...	$\mathbf{m_{s+1}}$	...	m_2

End of the construction.

Based on the construction, it is clear that $M_e = \cup_{s\in\omega} M_{e,s}$ is a computable magma. Moreover, the identity requirements is satisfied if and only if W_e stops accepting new elements eventually (i.e., W_e is finite). Thus, we showed that FIN is reducible to the index set of magmas with identity elements. And since this index set belongs to Σ_2^0 class, detecting an identity magma is Σ_2^0-complete within the class of computable magmas.

$\square$

Bibliography

[1] S.I. Adian, The unsolvability of certain algorithmic problems in the theory of groups, *Trudy Moskov. Mat. Obsc.* **6** (1957) 231–298.

[2] C.J. Ash and J.F. Knight, *Computable Structures and the Hyperarithmetical Hierarchy* (Elsevier, Amsterdam, 2000).

[3] I. Bilanovic, J. Chubb and S. Roven, Detecting properties from descriptions of groups, *in preparation*.

[4] R.N. Buttsworth, A family of groups with a countable infinity of full orders, *Bulletin of the Australian Mathematical Society* **4** (1971) 97–104.

[5] W. Calvert, V. Harizanov, J. Knight and S. Miller, Index Set of Computable Structures, *Algebra and Logic* **45** (2006) 306–325.

[6] J. Carson, V. Harizanov, J. Knight, K. Lange, C. Maher, C. McCoy, A. Morozov, S. Quinn and J. Wallbaum, Describing free groups, *Transactions of the American Mathematical Society* **364** (2012) 5715–5728.

[7] J. Chubb, *Ordered Structures and Computability*, PhD dissertation, George Washington University (2009).

[8] J. Chubb, M. Dabkowski and V. Harizanov, Groups with orderings of arbitrary algorithmic complexity, in *Sets and Computations*, (IMS, National University of Singapore, Lecture Notes Series 33), eds. R. Dilip, S.D. Friedman, and Y. Yang, World Scientific (2017) 221–251.

[9] S.B. Cooper, *Computability Theory*, Chapman Hall/CRC Mathematics Series, Chapman and Hall/CRC (2004).

[10] M.A. Dabkowska, *Turing Degree Spectra of Groups and Their Spaces of Orders*, PhD dissertation, George Washington University (2006).

[11] M.A. Dabkowska, M.K. Dabkowski, V.S. Harizanov, J.H. Przytycki and M.A. Veve, Compactness of the space of left orders, *Journal of Knot Theory and Its Ramifications* **16** (2007) 257–266.

[12] M.A. Dabkowska, M.K. Dabkowski, V.S. Harizanov and A. Togha, Spaces of orders and their Turing degree spectra, *Annals of Pure and Applied Logic* **161** (2010) 1134–1143.

[13] M. Dehn, Transformationen der Kurven auf zweiseitigen Flächen, *Mathematische Annalen* **71** (1911) 116–144.

[14] M. Dehn, Über unendliche diskontinuierliche Gruppen, *Mathematische Annalen* **72** (1912) 413–421.

[15] B. Deroin, A. Navas and C. Rivas, *Groups, orders, and dynamics*, arXiv preprint arXiv:1408.5805v2, 2016.

[16] V.P. Dobritsa, Some constructivizations of abelian groups, *Siberian Mathematical Journal* **24** (1983) 167–173 (English translation).

[17] R.G. Downey and S.A. Kurtz, Recursion theory and ordered groups, *Annals of Pure and Applied Logic* **32** (1986) 137–151.

[18] R. Fenn, C. Rourke, Racks and links in codimension two, *Journal of Knot Theory with Ramifications* **1(4)** (1992) 343–406.

[19] E. Fokina, V. Harizanov and A. Melnikov, Computable model theory, in *Turing's Legacy: Developments from Turing Ideas in Logic*, ed. R. Downey (Cambridge University Press/ASL, 2014) 124–194.

[20] A.M.W. Glass, *Partially Ordered Groups* (World Scientific, Singapore, 1999).

[21] V. Harizanov, Pure computable model theory, in *Handbook of Recursive Mathematics*, Vol. 1, eds. Yu.L. Ershov, S.S. Goncharov, A. Nerode, and J.B. Remmel (North-Holland, Amsterdam, 1998) 3–114.

[22] M. Harrison-Trainor, Left-orderable computable groups, *Journal of Symbolic Logic*, to appear.

[23] C.G. Jockusch and R.I. Soare, Π^0_1 classes and degrees of theories, *Transactions of the American Mathematical Society* **173** (1972) 33–56.

[24] C.G. Jockusch and R.I. Soare, Degrees of members of Π^0_1 classes, *Pacific Journal of Mathematics* **40** (1972) 605–616.

[25] D. Joyce, A classifying invariant of knots: the knot quandle, *Journal of Pure and Applied Algebra* **23** (1982) 37–65.

[26] S.C. Kleene, Recursive predicates and quantifiers, *Transactions of the American Mathematical Society* **53** (1943) 41–73.

[27] A.I. Kokorin and V.M. Kopytov, *Fully Ordered Groups* (Halsted Press, New York, 1974).

[28] V.M. Kopytov and N.Ya. Medvedev, *Right-Ordered Groups*, Siberian School of Algebra and Logic (Consultants Bureau, New York, 1996).

[29] P.A. Linnell, The space of left orders of a group is either finite or uncountable, *Bulletin of the London Mathematical Society* **43** (2011) 200–202.

[30] R.C. Lyndon and P.E. Schupp, *Combinatorial Group Theory*, Springer-Verlag, Berlin, 1977.

[31] S.V. Matveev, Distributive groupoids in knot theory, *Math. USSR-Sbornik*, **47** (1982) 73–83.

[32] S.H. McCleary, Free lattice-ordered groups represented as o-2 transitive l-permutation groups, *Transactions of the American Mathematical Society* **290** (1985) 69–79.

[33] C. McCoy and J. Wallbaum, Describing free groups, part II: Π^0_4 hardness and no Σ^0_2 basis, *Transactions of the American Mathematical Society* **364** (2012) 5729–5734.

[34] G. Metakides and A. Nerode, Effective content of field theory, *Annals of Mathematical Logic* **17** (1979) 289–320.

[35] A. Mostowski, On definable sets of positive integers, *Journal of Symbolic Logic* **13** (1948) 112–113.

[36] R.B. Mura and A. Rhemtulla, *Orderable Groups*, Lecture Notes in Pure and Applied Mathematics (Marcel Dekker, New York, 1977).

[37] A. Navas, On the dynamics of (left) orderable groups, *Annales de l'Institut Fourier* **60** (2008) 1685–1740.

[38] D. Rolfsen and B. Wiest, Free group automorphisms, invariant orderings, and topological applications, *Algebraic & Geometric Topology* **1** (2001) 311–320.

[39] A.S. Sikora, Topology on the spaces of orderings of groups, *Bulletin of the London Mathematical Society* **36** (2004) 519–526.

[40] R.I. Soare, *Turing Computability* (*Theory and Applications*) (Springer, Berlin, 2016).

[41] R.I. Soare, *Recursively Enumerable Sets and Degrees. A Study of Computable Functions and Computably Generated Sets (*Springer, Berlin, 1987).

[42] D.R. Solomon, Π^0_1 classes and orderable groups, *Annals of Pure and Applied Logic* **115** (2002) 279–302.

[43] D.R. Solomon, *Reverse Mathematics and Ordered Groups*, PhD dissertation, Cornell University (1998).

[44] A. Turing, Systems of logic based on ordinals, *Proc. London Mathematical Society* **45** (1939).