Undergraduate Research Opportunity Program
(UROP) Project Report

Semiautomatic Ordinal and Ring Structures

By

Qi Ji

Department of Computer Science

School of Computing

National University of Singapore

2018/2019

Undergraduate Research Opportunity Program
(UROP) Project Report

Semiautomatic Ordinal and Ring Structures

By

Qi Ji

Department of Computer Science

School of Computing

National University of Singapore

2018/2019

Project No: U107040
Advisor: Prof Frank Stephan
Deliverables:
Report: 1 Volume

Abstract

Semiautomatic structures generalise automatic structures in the sense that for some of the relations and functions in the structure one only requires the derived relations and structures are automatic when all but one input are filled with constants. One can also permit that this applies to equality in the structure so that only the sets of representatives equal to a given element of the structure are regular while equality itself is not an automatic relation on the domain of representatives. We look at semiautomatic rings with automatic addition and comparison and we also examine arithmetic on ordinals with semiautomatic multiplication.

Subject Descriptors:

F.4.1 Mathematical Logic

Keywords:

Automatic Structures

Acknowledgements

I would like to thank my advisor, professor Frank Stephan, for his invaluable advice and support. Thanks are also in order to my friends, especially Thomas Tan and Naïm Favier, for the many fruitful discussions that we had.

Contents

1 Introduction

The study of automatic structures was initiated by the pioneering works of Hodgson [Hod76; Hod83], Khoussainov and Nerode [KN94] and Blumensath and Grädel [BG00]. In mathematics and computer science, it is of interest to classify structures in which operations are computationally easy, with a low computational complexity.

For the various characterisations of automatic/regular sets we refer the reader to resources on theory of computation such as [HMU06] and [Ste18].

Description 1. Relations over a base set $A \subseteq \Sigma^*$ are usually encoded as subsets of A^n, where n is the arity of our relation. Given multiple inputs, we use the standard method of encoding several strings synchronously. Let $a = a_0a_1 \cdots a_n$ and $b = b_0b_1 \cdots b_m$ in A, the convolution $conv(a, b)$ is defined as $c_0c_1 \cdots c_{\max(m,n)}$ where

- $c_k = \binom{a_k}{b_k}$ if $k \leq m$ and $k \leq n$,
- $c_k = \binom{a_k}{\#}$ if $m < k \leq n$, and
- $c_k = \binom{\#}{b_k}$ if $n < k \leq m$,

with $\#$ being a fixed padding character not in Σ. This process naturally identifies $A \times A$ with the set of all convolutions $\{conv(x, y) : x, y \in A\}$ and is easily generalised to arities greater than 2.

A relation $R \subseteq A^n$ is automatic if its representation as indicated is a regular set, while a function $f : A^n \to A$ is automatic if its graph (as a subset of A^{n+1}) is regular.

1.1 Automata on group-like structures

Groups, semigroups and monoids are characterised by having a single binary operation over a base set [Bou07]. Groups have been studied in mathematics for centuries in their own right, but only recently have the connection between group theory and automata theory been made [Hod76; Hod83; KN94; Eps+92].

We first introduce the framework proposed by Hodgson [Hod76; Hod83] and Khoussainov and Nerode [KN94].

Definition 2. We call a semigroup $(G, \circ)$ **fully automatic** iff

- G is regular over Σ^* where Σ is a finite alphabet,
- $\circ : G \times G \to G$ is an automatic function.

Additionally, if $(H, \star)$ is isomorphic to $(G, \circ)$ we call H fully automatic too. Fully automatic monoids and groups are defined analogously.

Remark. In the original literature, the authors referred to this definition as just "automatic". We follow the convention in [Ste18] and use the term "fully automatic" in the sense that the *full* semigroup operation is automatic. At the same time, we also disambiguate it from the definitions introduced below.

In [Eps+92], the authors argued that the formalisation in Definition 2 is, from the point of view of finitely-generated groups, too restrictive, and proposed the following definition.

Definition 3. Let $(G, \circ)$ be a semigroup generated by a finite subset $F \subseteq G$. The semigroup $(G, \circ)$ is **automatic** iff

- G is a regular subset of F^*,
- each $x \in G$ has a unique representative in F^*, and
- for each $y \in G$, the multiplication map $(\circ y) : G \to G$ defined as $x \mapsto x \circ y$ is automatic.

By representing elements as words over generators, the representatives are more meaningful. Note that this definition is, in the case of finitely-generated groups, weaker than Definition 2, as the example illustrates.

Example 4. Consider the semigroup $(\Delta^*, \circ)$ where $2 \leq |\Delta| < \infty$ and $\circ$ denotes concatenation of words. We can verify that this semigroup satisfies Definition 3 as for each $y \in \Delta^*$, the map $x \mapsto xy$ is automatic. This semigroup fails to satisfy Definition 2 however, as in order to recognise $\{conv(x, y, z) : xy = z\}$ using a synchronous finite automata, we have to store y as we are reading in x and comparing it with z. The problem is that the length of y is unbounded, so the graph of $\circ$ cannot be recognised with finitely many states. In fact this group has no fully automatic representation.

Kharlampovich, Khoussainov and Miasnikov were the first to formally consider the related concept of a Cayley automatic group [KKM11]. It is sometimes also called graph automatic because it considers the Cayley graph of a group as the automatic structure.

Definition 5. A finitely generated group G generated by F is **Cayley automatic** iff the following conditions hold for some finite alphabet Σ,

- representatives of G form a regular subset H of Σ^*,
- each $x \in G$ has a unique representative in H,
- for each $y \in F$, the right multiplication by y map is an automatic map $H \to H$.

This is a generalisation of Definition 3, where we drop the condition that natural representatives are chosen, that is $\Sigma = F$ the set of generators. This gives us a even bigger class of automatic groups.

Example 6. The Heisenberg group $\mathcal{H}_3(\mathbb{Z})$ defined as

$$\left\{ \begin{pmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{pmatrix} : a, b, c \in \mathbb{Z} \right\}$$

is Cayley automatic (6.6 in [KKM11]), but not automatic (8.1.1 in [Eps+92]).

The low complexity of automata motivates the search for automatic presentations of various algebraic structures. For example, the word problem for groups in general is well-known to be undecidable. However, most reasonable formalisations of automatic groups will have their word problem solvable with a quadratic time complexity [Eps+92; KKM11].

1.2 General automatic structures

For general mathematical structures $(A, R_1, R_2, \dots, R_m, f_1, f_2, \dots, f_n)$, where A is a base set, each R_i is a relation over A and f_j a finitary function $A^k \to A$, we consider the more general framework of Hodgson [Hod76; Hod83] and Khoussainov and Nerode [KN94].

Definition 7. A structure $(A, R_1, R_2, \dots, R_m, f_1, f_2, \dots, f_n)$ is automatic iff

- the set A is regular in Σ^* where Σ is some finite alphabet,
- all the relations $R_1, R_2, \dots, R_m$ are automatic, and
- all the functions $f_1, f_2, \dots, f_n$ are automatic.

In the spirit of Definition 2, we do not distinguish between structures that are automatic and structures that are merely isomorphic to an automatic structure.

Example 8. A fully automatic group $(G, \varepsilon, \circ)$ satisfying Definition 2 is an automatic structure.

Automatic structures are of mathematical interest due to their low computational complexity and closure under first-order definability, as shown in [KN94].

Theorem 9. *Any functions and relations that are definable using a formula of first-order in terms of automatic functions and relations is again automatic. Furthermore there is an effective procedure to construct the resultant automata from that used in the parameters to define the function or relation.*

1.3 Semiautomatic structures

Seeking more general ways to utilise finite automata for representing non-automatic structures, Jain, Khoussainov, Stephan, Teng and Zou proposed semiautomatic structures as a generalisation of Definition 7 [Jai+17].

Definition 10. Let $f : R^n \to R$ be a function. f is **semiautomatic** iff fixing $n-1$ inputs, the resultant $R \to R$ function is automatic. The definition of a semiautomatic relation is analogous.

Definition 11. A structure $(A, f_1, f_2, \dots, f_m, R_1, R_2, \dots, R_n; g_1, g_2, \dots, g_p, S_1, S_2, \dots, S_q)$ is **semiautomatic** iff

- the set A is regular in Σ^* where Σ is some finite alphabet,
- all the functions and relations before the semicolon $f_1, f_2, \dots, f_m, R_1, R_2, \dots, R_n$ are automatic, and
- all the functions and relations after the semicolon $g_1, g_2, \dots, g_p, S_1, S_2, \dots, S_q$ are semiautomatic.

1.4 Ordinals

For readers familiar with set theory, this section can be skipped without loss of continuity.

Definition 12. A set α is an ordinal iff every $\beta \in \alpha$ is a subset of α and $(\alpha, \in)$ is a well order.

Ordinals can be thought as equivalence classes of well ordered sets. They naturally describe how many times a process is iterated, possibly transfinitely many times. The class of all ordinals is also well ordered by membership, and whenever α, β are ordinals, $\alpha < \beta$ denotes $\alpha \in \beta$.

Description 13 (Ordinal arithmetic)**.**

- The sum of two ordinals $\alpha + \beta$ expresses the order type of α placed before β, which is defined as the ordinal order-isomorphic to $(\{0\} \times \alpha \cup \{1\} \times \beta, \prec)$ equipped with dictionary ordering.
- The product of two ordinals $\alpha \cdot \beta$ can be thought of as β copies of α, which is defined as the ordinal order-isomorphic to the set $\beta \times \alpha$ equipped with dictionary ordering.
- Ordinal exponentiation α^β is repeated multiplication, and is defined as α multiplied by itself β many times.

We note that the initial segment containing the first ω many ordinals is exactly the natural numbers, which add and multiply just like natural numbers, therefore we sometimes use ω to refer to the set of natural numbers.

For a comprehensive text on axiomatic set theory, we refer the reader to [Kun80].

2 Semiautomatic ordinal structures

2.1 Representations with automatic addition

Delhommé proved the following characterisation of automatic ordinals in [Del04].

Theorem 14 (Delhommé)**.** *Let α be an ordinal, $(\alpha, +, <)$ is automatic iff $\alpha < \omega^\omega$. Here the domain of $+$ is the set of all pairs (β, γ) with $\beta + \gamma < \alpha$.*

We illustrate how any ordinal $\alpha < \omega^\omega$ is automatic with an example.

Example 15. The structure $(\omega^3, +, <)$ is automatic. Any ordinal below ω^3 is of the form $\omega^2 \cdot c_2 + \omega \cdot c_1 + c_0$ with $c_0, c_1, c_2 \in \mathbb{N}$. We can express any ordinal $\alpha = \omega^2 \cdot a_2 + \omega \cdot a_1 + a_0 < \omega^3$ as $conv(a_0, a_1, a_2)$. Consider $\alpha, \beta \in \omega^3$, where $\alpha = \omega^2 \cdot a_2 + \omega \cdot a_1 + a_0$ and $\beta = \omega^2 \cdot b_2 + \omega \cdot b_1 + b_0$. The sum $\alpha + \beta$ can be given by

$$\alpha + \beta = \begin{cases} \omega^2 \cdot (a_2 + b_2) + \omega \cdot b_1 + b_0 & \text{if } b_2 > 0 \\ \omega^2 \cdot a_2 + \omega \cdot (a_1 + b_1) + b_0 & \text{if } b_2 = 0, b_1 > 0 \\ \omega^2 \cdot a_2 + \omega \cdot a_1 + (a_0 + b_0) & \text{if } b_2 = 0, b_1 = 0 \end{cases}$$

because for any $n, m \in \mathbb{N}$ with $n > m$, we have $\omega^m + \omega^n = \omega^n$. This means we can define addition on our representatives using the expression given above as

$$conv(a_0, a_1, a_2) + conv(b_0, b_1, b_2) = \begin{cases} conv(b_0, b_1, a_2 + b_2) & \text{if } b_2 > 0 \\ conv(b_0, a_1 + b_1, a_2) & \text{if } b_2 = 0, b_1 > 0 \\ conv(a_0 + b_0, a_1, a_2) & \text{if } b_2 = 0, b_1 = 0 \end{cases}$$

and since there exists an automatic representation of $(\mathbb{N}, +, <)$, the addition function is in fact automatic, as it only performs checks that are automatic in our presentation of $\mathbb{N}$. Because comparison of ordinals below ω^3 is first-order definable in terms of addition, $<$ is an automatic relation.

Any $\alpha < \omega^\omega$ by definition of ordinal exponentiation is bounded above by ω^n for some

$n \in \omega$. Since Example 15 generalises to n naturally, we consider the natural embedding of $(\alpha, +, <)$ in $(\omega^n, +, <)$.

Theorem 14 came before the notion of semiautomatic structures was invented. Our main result here is that using the same representation, we get semiautomatic multiplication without losing automaticity of the other operations.

Theorem 16. *For any $\alpha < \omega^\omega$, the structure $(\alpha, +, <, =; \cdot)$ is semiautomatic. Similarly to Theorem 14 we only consider the domain of $+$ to be all pairs (β, γ) with $\beta + \gamma < \alpha$ and the domain of $\cdot$ to be all pairs (β, γ) with $\beta \cdot \gamma < \alpha$.*

Remark. As there is no automatic representation of $(\omega, \cdot, =)$, in general, for any infinite ordinal we cannot move $\cdot$ left of the semicolon.

Again without loss of generality we consider $\alpha = \omega^n$ for some $n \in \omega$. We use a representation similar to that illustrated in Example 15.

2.1.1 Left multiplication

Lemma 17. *Using the representation, for any fixed $\beta \in \omega^n$ the map $\gamma \mapsto \beta \cdot \gamma$ restricted to all γ satisfying $\beta \cdot \gamma < \omega^n$ is automatic.*

Proof. First express the ordinals in normal form

$$\begin{aligned}\beta &= \omega^k \cdot b_k + \omega^{k-1} \cdot b_{k-1} + \cdots + \omega \cdot b_1 + b_0 \\ \gamma &= \omega^l \cdot c_l + \omega^{l-1} \cdot c_{l-1} + \cdots + \omega \cdot c_1 + c_0\end{aligned}$$

where $b_k, c_l > 0$. We then make the following general observation that

$$\begin{aligned}\beta \cdot \gamma &= \beta \cdot \omega^l \cdot c_l + \beta \cdot \omega^{l-1} \cdot c_{l-1} + \cdots + \beta \cdot \omega \cdot c_1 + \beta \cdot c_0 \\ &= \left(\omega^k \cdot b_k + \omega^{k-1} \cdot b_{k-1} + \cdots + \omega \cdot b_1 + b_0\right) \cdot \omega^l \cdot c_l \\ &\quad + \left(\omega^k \cdot b_k + \omega^{k-1} \cdot b_{k-1} + \cdots + \omega \cdot b_1 + b_0\right) \cdot \omega^{l-1} \cdot c_{l-1} \\ &\quad + \ldots \\ &\quad + \left(\omega^k \cdot b_k + \omega^{k-1} \cdot b_{k-1} + \cdots + \omega \cdot b_1 + b_0\right) \cdot \omega \cdot c_1 \\ &\quad + \left(\omega^k \cdot b_k + \omega^{k-1} \cdot b_{k-1} + \cdots + \omega \cdot b_1 + b_0\right) \cdot c_0 \\ &= \omega^{k+l} \cdot c_l + \omega^{k+l-1} \cdot c_{l-1} + \cdots + \omega^{k+1} \cdot c_1 \\ &\quad + \left(\omega^k \cdot (b_k \cdot c_0) + \omega^{k-1} \cdot b_{k-1} + \cdots + \omega^{b_1} + b_0\right) \cdot 1_{c_0 \neq 0}\end{aligned}$$

where $1_{c_0 \neq 0}$ is 1 is $c_0 \neq 0$ and 0 otherwise.

If our assumptions on the domain of γ holds, then $k + l < n$. We have

$$\begin{aligned}
&\beta \cdot conv(c_0, c_1, \dots, c_l, 0, \dots, 0) \\
&= conv(\overbrace{0, \dots, 0}^{k+1 \text{ many}}, c_1, c_2, \dots, c_l, 0, \dots, 0) \\
&\quad + \begin{cases} 0 & \text{if } c_0 = 0 \\ conv(b_0, b_1, \dots, b_k - 1, b_k \cdot c_0, 0, \dots, 0) & \text{otherwise} \end{cases}
\end{aligned}$$

We see that this function is automatic as β is fixed so each b_i can be treated as constant, therefore computing $b_k \cdot c_0$ given c_0 is automatic, in addition the checks are also automatic and the final addition is also automatic due to Theorem 14. □

2.1.2 Right multiplication

We fix $\gamma = \omega^l \cdot c_l + \cdots + \omega \cdot c_1 + c_0$ an ordinal in normal form, then for any β,

$$\beta \cdot \gamma = \beta \cdot \omega^l \cdot c_l + \cdots + \beta \cdot \omega \cdot c_1 + \beta \cdot c_0.$$

Hence we can express right-multiplication by γ as a finite composition of the following

- right-multiplication by ω,
- right-multiplication by fixed constants $c_0, c_1, \dots, c_l$,
- ordinal additions.

For we are using a representation of ω^n where Theorem 14 holds, ordinal addition and right-multiplication by fixed constants (implemented as repeated addition) is automatic, and we are reduced to showing the following.

Lemma 18. *Using the same representation, the map $\beta \mapsto \beta \cdot \omega$ restricted to all β satisfying $\beta \cdot \omega < \omega^n$ is automatic.*

Proof. For simplicity we demonstrate with the case $n = 4$. Let $\beta = \omega^3 \cdot b_3 + \omega^2 \cdot b_2 + \omega \cdot b_1 + b_0$

be in normal form, using the rules of ordinal multiplication we have

$$\begin{aligned}\beta \cdot \omega &= (\omega^3 \cdot b_3 + \omega^2 \cdot b_2 + \omega \cdot b_1 + b_0) \cdot \omega \\ &= \begin{cases} \omega^4 & \text{if } b_3 > 0 \\ \omega^3 & \text{if } b_3 = 0, b_2 > 0 \\ \omega^2 & \text{if } b_3 = 0, b_2 = 0, b_1 > 0 \\ \omega & \text{if } b_3 = 0, b_2 = 0, b_1 = 0, b_0 > 0 \\ 0 & \text{otherwise .} \end{cases}\end{aligned}$$

If our assumptions on the domain of β holds, then $b_3 = 0$, and we have

$$conv(b_0, b_1, b_2, 0) \cdot \omega = \begin{cases} conv(0,0,0,1) & \text{if } b_2 > 0 \\ conv(0,0,1,0) & \text{if } b_2 = 0, b_1 > 0 \\ conv(0,1,0,0) & \text{if } b_2 = 0, b_1 = 0, b_0 > 0 \\ conv(0,0,0,0) & \text{otherwise} \end{cases}$$

where only automatic checks on the input is used. □

Since multiplication on both sides can be made semiautomatic without losing what we already have in this representation, Theorem 16 follows.

2.2 Representations with semiautomatic addition

By Theorem 14, any ordinal $\alpha \geq \omega^\omega$ cannot admit automatic addition and comparison, but by allowing our operations to be semiautomatic, we get more ordinals. Our main goal in this section would be to show that we can get a representation of ω^ω in which addition, multiplication and equality are simultaneously semiautomatic.

Theorem 19. *The structure* $(\omega^\omega; +, <, \cdot, =)$ *is semiautomatic.*

Instead of tackling this straight on, we present a related result.

2.2.1 Polynomials over $\mathbb{N}$

This result was only sketched as part of Theorem 37 in [Jai+17], where the authors presented a more general construction.

Theorem 20. *The semiring of polynomials over* $\mathbb{N}$ $(\mathbb{N}[x]; +, \cdot, =)$ *is semiautomatic.*

Proof. We let $A \subseteq \Sigma^*$ be an semiautomatic representation of $(\mathbb{N}, +, <; \cdot)$ and let $\oplus, \otimes, \bullet$ be symbols outside Σ.

We represent elements of $\mathbb{N}[x]$ as strings in $B = \{\varepsilon\} \cup ((A \cdot \{\bullet\})^* \cdot A)$, where we encode nonzero polynomials like $c_n x^n + \cdots + c_1 x + c_0$ as $c_0 \bullet c_1 \bullet \ldots \bullet c_n$, where $c_n \neq 0$.

For each string $w \in C = \{\varepsilon\} \cup ((B \cdot \{\oplus, \otimes\})^* \cdot B)$ we assign a value $val(w)$ in B as follows

- $val(\varepsilon)$ is 0 which is represented by ε;
- For $w \in B$, $val(w)$ is the base element with trailing zeroes stripped;
- $val(w \oplus \varepsilon) = val(w)$;
- $val(w \otimes \varepsilon)$ is 0 which is represented by ε;
- $val(w \oplus v) = val(w) + val(v)$ when $v \in B$;
- $val(w \otimes v) = val(w) \cdot val(v)$ when $v \in B$ and v does not represent 0.

Note that val is not automatic, but we can define val_n such that $val_n(w) = val(w)$ for enough w. For each natural number n, $w \in C$, define $val_n(w) = val(w)$ if w represents a polynomial of degree n or less and all coefficients are below n. In the other case $val_n(w) = @$ if the degree of w is above n or some coefficient greater or equal to n is encountered. Notice that val_n is automatic as whenever its value goes to @, it remains at @ until a multiplication with zero occurs. In any other case the automaton only has to handle finitely many possible values. The finiteness property in here still holds because arithmetic in $\mathbb{N}$ has neither additive inverses nor zero divisors.

For $v = \sum_{i \leq \deg(v)} c_i x^i$ we define $d(v) = \max(\deg(v), c_i)$. The rest of the argument is similar to that of Theorem 37 in [Jai+17]. We can check if $w \in C$ is equal to a fixed value v by comparing $val_{d(w)}(w)$ with v. We can also do addition with a fixed v as follows. If $val_{d(v)}(w) \neq @$ then we represent $v + w$ using the result, else we use $v \oplus w$. If $v \neq \varepsilon$ the product $v \cdot w$ are represented by ε, otherwise we represent the product with $v \otimes w$. □

2.2.2 Going back to ω^ω

By thinking of ordinals below ω^ω as natural polynomials "evaluated" at ω, the corresponding proof for Theorem 19 is mostly similar to that of Theorem 20, but we need to make a few concessions.

As ordinal addition and multiplication are not commutative, we have to introduce more symbols $\oplus_l, \oplus_r, \otimes_l, \otimes_r$ and expand the definition of val accordingly. So for example $val(v \oplus_l w) = val(v) + val(w)$ and $val(v \oplus_r w) = val(w) + val(v)$.

When adding and multiplying ordinals, arbitrarily large coefficients could disappear. For instance, $k + \omega = \omega$ and $\omega \cdot k \cdot \omega = \omega \cdot \omega$ no matter how large $k \in \mathbb{N}$ is. So in the definition of val_n, we require more error conditions. Where our polynomial coefficients were denoted using a semiautomatic representation of $(\mathbb{N}, +, <; \cdot)$, we add a new symbol $\odot$ which denotes "coefficient too large", and when the result of an operation goes beyond n we output $\odot$ in place of the coefficient. Then the existing error case @ is relegated to when the degree of the result becomes too large.

To see that comparison can also be made semiautomatic, fix $\beta < \omega^\omega$, to see if $\alpha \in \omega^\omega$ satisfies $\alpha < \beta$, we can look at $val_{d(\beta)}(\alpha)$ and examine the coefficients.

3 Semiautomatic ring structures

Rings are obtained by adding to an Abelian group a notion of multiplication.

Definition 21. $(R, +, \cdot, 0, 1)$ is a ring if $(R, +, 0)$ is an abelian group, $(R \setminus \{0\}, \cdot, 1)$ is a monoid and $\cdot$ distributes over $+$.

3.1 Golden ratio

In [Jai+17] the authors showed that the ring $(\mathbb{Z}(\sqrt{n}), \mathbb{Z}, +, <, =; \cdot)$ for every $n \in \mathbb{N}$ has a semiautomatic presentation. The structure is best illustrated using a simple irrational, so in this section u denotes the Golden Ratio $\frac{1+\sqrt{5}}{2}$.

Theorem 22. *The ordered ring $(\mathbb{Z}[u], +, <, =; \cdot)$ has a semiautomatic presentation.*

3.1.1 Isolating finiteness

Note that $3 = u^{-2} + u^2$ and 3 dominates all other coefficients, so given any $a = p + qu \in \mathbb{Z}[u]$, with repeated applications of this identity, we can express a as a longer (but still finite) linear combination of powers of u

$$a = \sum_i a_i u^i$$

where each $|a_i| \leq 2$.

3.1.2 The representation

Appealing to earlier part, we represent $a \in \mathbb{Z}[u]$ as a sequence of integers $a_n a_{n-1} \cdots a_m$ satisfying

- $n \geq 0 \geq m$,
- $a = \sum_i a_i u^i$ with each $|a_i| \leq 2$.

Since $u^{-1} = u - 1$, we are not representing extra elements.

3.1.3 Tail bounds

Note that the geometric series $\sum_{i\leq 2} u^i$ converges to $\frac{u^3}{u-1}$, so in particular the tail sum as i decreases is bounded. Now let $k' \in \mathbb{N}$ such that $k' \geq 2 \cdot 2 \cdot \sum_{i\leq 2} u^i$. This allows us to come up with a sign test algorithm for our representatives.

3.1.4 Sign test

Let $a = \sum_{i=m,\ldots,0,\ldots,n} a_i u^i \in \mathbb{Z}[u]$, the following pseudocode tests if $a > 0, a = 0$ or $a < 0$.

Algorithm 1: Sliding over coefficients to test sign

```
initialise x, y ← 0, i ← n
while i ≥ m do
    a_{i+1} ← a_{i+1} + a_{i+2}
    a_i ← a_i + a_{i+2}
    a_{i+2} ← 0
    if a_{i+1} > k' ∨ a_{i+2} > k' then
        halt with a > 0
    if a_{i+1} < −k' ∨ a_{i+2} < −k' then
        halt with a < 0
    i ← i − 1
determine sign by a finite case distinction over the values of x and y
```

We basically go through coefficients a_i with a sliding window of size 2, updating using the relation $u^2 = u + 1$. By our tail bounds, the early termination always outputs the correct answer. Furthermore by appealing to the characterisation of automatic functions as those computable by linear-time one-tape Turing machines where input and output start at the same position [Cas+12], this algorithm is automatic.

With an automatic sign test primitive, it is easy to see that addition, comparison and equality could be implemented in an automatic matter. Furthermore, multiplication by any power of u can be realised by shifting all coefficients while multiplication with a fixed integer can be implemented as repeated addition. Multiplying by any member in $\mathbb{Z}[u]$ can be implemented as a finite composition of such operations, which gives semiautomatic multiplication.

In [Jai+17] the authors naturally generalised this observation to get semiautomatic representations of $(\mathbb{Z}(\sqrt{n}), \mathbb{Z}, +, <, =; \cdot)$ for any square root $\sqrt{n}$, the details of which we omit. We do note that the generalisation heavily hinges on the fact that for any $n \in \mathbb{N}$ that is not a perfect square, its associated Pell's equation $d^2 - ne^2 = 1$ has infinitely

many solutions [Lag67].

3.2 Cube roots

Trying to generalise this observation for cube roots is not trivial, as we are still not aware of any deeper understanding of the cubic rings $\mathbb{Z}[\sqrt[3]{n}]$ in general. However we do have some preliminary results.

Example 23. There is a semiautomatic ring $(A, +, =, <; \cdot)$ containing $\sqrt[3]{7}$.

We choose

$$u^{-1} = 2 - \sqrt[3]{7}$$

and consider the semiautomatic ring $(\mathbb{Z}[u], +, <, =; \cdot)$, then we have the relation $1 - 12u^{-1} + 6u^{-2} - u^{-3} = 0$, where 12 dominates the sum of all other terms, allowing us to use a representation similar to that for the Golden ratio case, with coefficients bounded between -12 and 12.

Note that $u^{-1} = u^2 - 12u + 6$ so we are still not representing extra elements.

When checking if $a + b = c$, we perform the sign test on $a + b - c$, so assume the coefficients are between -36 and 36. This time, we perform the sign test using a sliding window of size 3, as we use the relation $u^3 = 12u^2 - 6u + 1$ to update our variables.

For a tail bound, we have

$$\sum_{i \leq 1} u^i \leq \frac{1}{10}.$$

Instead of using a flat bound k', we terminate when one of the following conditions is broken

- $|a_{i+2}| \leq 16\hat{c}$,
- $|a_{i+1}| \leq 4\hat{c}$, or
- $|a_i| \leq \hat{c}$.

where $\hat{c} = 360$. The reader can verify that violating one of the conditions will conclusively indicate that the number exceeds the tail sum in a specific direction, which allows us to determine its sign.

4 Conclusion

Jain, Khoussainov, Stephan, Teng and Zou [Jai+17] studied semiautomatic structures. In particular, they presented semiautomatic ordered rings where addition, subtraction, order and equality are in fact automatic. They also investigated countable ordinals with semiautomatic addition, ordering and equality. We have investigated natural extensions to some of the problems, such as considering multiplicative ordinal structures, and extending their results on quadratic rings to cubic rings.

It seems fully possible to extend the ideas for Example 23 into other cube roots. The challenge here is that cubic rings $\mathbb{Z}[\sqrt[3]{n}]$ is generally less well-understood, and significant insights might require breakthroughs in number theory.

Also, it seems plausible for larger ordinals to admit semiautomatic addition, multiplication and equality, not just ω^ω as in Theorem 19.

References

[BG00] A. Blumensath and E. Gradel. 'Automatic structures'. In: *Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No.99CB36332)*. June 2000, pp. 51–62.

[Bou07] Nicolas Bourbaki. *Algèbre*. Berlin New York: Springer, 2007.

[Cas+12] John Case, Sanjay Jain, Samuel Seah and Frank Stephan. 'Automatic Functions, Linear Time and Learning'. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 96–106.

[Del04] Christian Delhommé. 'Automaticité des ordinaux et des graphes homogènes'. In: *Comptes rendus - Mathématique* 339.1 (2004), pp. 5–10.

[Eps+92] David B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. Levy, M. S. Paterson and W. P. Thurston. *Word Processing in Groups*. Natick, MA, USA: A. K. Peters, Ltd., 1992.

[HMU06] John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006.

[Hod76] Bernard Ralph Hodgson. 'Théories décidables par automate fini'. PhD thesis. Département de mathématiques et de statistique, Université de Montréal, 1976.

[Hod83] Bernard Ralph Hodgson. 'Décidabilité par automate fini'. In: *Annales des sciences mathématiques du Québec* 7.1 (1983), pp. 39–57.

[Jai+17] Sanjay Jain, Bakhadyr Khoussainov, Frank Stephan, Dan Teng and Siyuan Zou. 'Semiautomatic structures'. In: *Theory of Computing Systems* 61.4 (2017), pp. 1254–1287.

[KKM11] Olga Kharlampovich, Bakhadyr Khoussainov and Alexei Miasnikov. 'From automatic structures to automatic groups'. In: *Groups, Geometry, and Dynamics* 8 (1 July 2011).

[KN94] Bakhadyr Khoussainov and Anil Nerode. 'Automatic Presentations of Structures'. In: *Logical and Computational Complexity. Selected Papers. Logic and Computational Complexity, International Workshop LCC '94, Indianapolis, Indiana, USA, 13-16 October 1994*. Ed. by Daniel Leivant. Vol. 960. Lecture Notes in Computer Science. Springer, 1994, pp. 367–392.

[Kun80] Kenneth Kunen. *Set Theory: An Introduction to Independence Proofs*. North-Holland, 1980.

[Lag67] Joseph-Louis Lagrange. 'Solution d'un probléme d'Arithmtique'. In: *Oeuvres de Lagrange* 1 (1867). Ed. by Joseph Alfred Serret, pp. 671–731.

[Ste18] Frank Stephan. 'Methods and theory of automata and languages'. In: *Lecture Notes, School of Computing, National University of Singapore* (2018).