Automatic maps on the Gaussian integers

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfillment of the requirements
for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

T.J.P. Krebs
Delft, the Netherlands
September 2013

Copyright © 2013 by T.J.P. Krebs. All rights reserved.
Contents

1 Introduction 5

2 Formal languages and automata 6
   2.1 Formal Languages .................................................. 6
   2.2 Automata .............................................................. 6
   2.3 Regular Languages .................................................. 11

3 Numeration systems for the Gaussian integers 13
   3.1 Gaussian integers .................................................. 13
   3.2 Numeration systems ................................................. 16
   3.3 Numeration systems for the Gaussian integers .................. 18

4 Automatic maps on the Gaussian integers 27
   4.1 General definition and basic properties .......................... 27
   4.2 Multiplicative independence of bases ............................. 35
1 Introduction

The motivation for this work originates from the question how Cobham’s deep theorem for automatic sequences can be extended to work for automatic maps on the Gaussian integers. In the ordinary case of automatic sequences, Cobham’s theorem fully characterizes the bases in which a particular sequence is automatic. The formulation of an analogue for the Gaussian integers is not known (see [DM10]), however, in [HS03] a conjecture is made to relate \((-a + i)-\text{automaticity}\) to \((-b + i)-\text{automaticity}\), where \(a\) and \(b\) are positive integers. But what about other Gaussian bases?

That touches upon a subproblem of the posed question that is the subject of this work. How can you define automatic maps on the Gaussian integers? The two key components of an automatic map are an automaton (Chapter 2), and a numeration system that represents every Gaussian integer exactly once (Chapter 3). However, the literature is scarce on this latter subject and mostly provides the system of [KS75] in base \(-b + i\) and digits \(\{0, 1, \ldots, b^2\}\) that is used by [HS03], and the famous revolving system of [DK70] in base \(1 + i\) and digits \(\{0, 1, -i, -1, i\}\). But there is also mention of a result that gives almost everything we are after. In [DDG78] it is shown among other things that a numeration system exists for the Gaussian integers in every base with modulus more than 2 by exploring digit sets other than \(\{0, 1, \ldots, b^2\}\). Unfortunately, the report proved hard to find so we reproduce the result here. Finally, we wanted to support the revolving system as well, so we chose to loosen the restriction that every Gaussian integer is represented exactly once to at least once.

Merely establishing the existence of a numeration system for the Gaussian integers in every base is not enough though. Another aspect of defining automatic maps is the particular choice of numeration system (Chapter 4). Somehow this does not really come up when working with automatic sequences since the choice of digits and representations seems very natural. We would like to have our concept of automatic maps to be well-defined in the sense that it does not matter which numeration system in a given base we pick, and we show that this indeed holds true. We then go on to prove that a map is automatic with respect to every multiplicatively dependent base, and show that there exist automatic maps that are not automatic in any multiplicatively independent base. Consequently, it reveals partly how an analogue of Cobham’s theorem for the Gaussian integers will look like, and answers the open question in [ACG+97] negatively.

The results given in section 4.1 are highly comparable to those presented in [ACG+97]. Since it was not hard to state most of our results for arbitrary commutative rings (with the notable exception of studying automaticity in multiplicatively independent bases), we have done so to allow for a better comparison. The main difference is that we are less restrictive in the choice of numeration system, allowing us to support automatic maps in all small Gaussian bases.
2 Formal languages and automata

In this chapter we will briefly go over the constructs and properties of formal languages and finite automata. The material here can be found in any introductory text to these subjects, for instance \[\text{[Sip06]}\].

2.1 Formal Languages

The most basic parts of a language are letters (or symbols) like \(a\) or \(b\), which we will display in a typewriter font. A non-empty finite collection of letters such as \(\{a, b\}\) is called an alphabet. A word is a finite or infinite sequence of letters, e.g. \((b, a, a)\), and we will use \(\epsilon\) to denote the empty word. A language is a collection of finite words over some alphabet. An example of a language would be the collection of all words over \(\{a, b\}\) that are palindromes, which begins like \(\{\epsilon, (a), (b), (a, a), (b, b), (a, a, a), (a, b, a), \ldots\}\).

We will supply the structures we just introduced with some useful operations.

Operations on letters or alphabets There is no need to define any operation on letters or alphabets explicitly. Instead, when operating on letters we treat letters as single-element words. In particular, this means that we treat alphabets as languages.

Operations on words As with any sequence, we may index words (starting from 0), take lengths and take subsequences. The length of a finite word \(w\) is written as \(|w|\). A contiguous subsequence is also known as a subword. The subword starting at the first (ending at the last) letter of a word is a prefix (suffix).

A fundamental operation on words is concatenation. We can concatenate a finite word \(v\) and a word \(w\) by juxtaposing their letters, resulting in \(v \circ w\). For instance, \((b, a) \circ a = (b, a, a)\). We denote this operation multiplicatively with the usual notational benefits, meaning we may omit the operator and let \(w^n\) be short for \(ww \ldots w\) (\(n\) times). Since concatenation is associative, there is also no need for parentheses to specify the order of operations. The word \((b, a, a)\) can therefore be written succinctly as \(ba^2\).

Another operation is the reversal of words. The reversal \(w^R\) of a finite word \(w\) of length \(n\) is the word \(w_{n-1} \ldots w_1 w_0\), so a palindrome is a word \(w\) satisfying \(w = w^R\).

Operations on languages On top of the usual set-theoretic operations, we lift any operation on words to operate on languages by element-wise application on the Cartesian product of its operands. For instance, \(\{a, ab\}^R = \{a, ba\}\) and \(\{a, b\} \circ \{a, b\} = \{aa, ab, ba, bb\}\). We will favor concatenation over the Cartesian product as the default product on languages, so the previous example can be shortened to \(\{a, b\}^2\).

Additionally, we define for any language \(L\) both the Kleene star operation \(L^* = \bigcup_{n \in \mathbb{N}} L^n\), and the Kleene plus operation \(L^+ = LL^*\). In this work we will take \(\mathbb{N}\) to include 0, so keep in mind that \(L^0 = \{\epsilon\}\). We have for example that \(\{a, b\}^* = \{\epsilon, a, b, aa, ab, ba, \ldots\}\) represents the set of all finite words over the alphabet \(\{a, b\}\). A language is simply a subset of \(\Sigma^*\) for some alphabet \(\Sigma\).

2.2 Automata

An automaton is an abstract, state-based machine that processes words, and is one of the simplest models of computation. In essence, an automaton is a labeled graph on which it

\[\footnote{A palindrome is a word that reads the same forward as backward.}\]
traverses a path labeled by input symbols starting at a designated state (or vertex). See for instance figure 1 for a particular type of automaton. The initial state is indicated by the incoming arrow without a source, here $q_{bb}$. If we feed the depicted automaton the word $babaa$ letter by letter, then it will successively visit the states $q_{bb}$, $q_{ba}$, $q_{ab}$, $q_{ba}$, and finally $q_{aa}$. After having consumed the entire input word, it will make a decision whether to accept or reject the input based on the state it lands in. The accepting states are marked with a double circle, so we see that the example automaton accepts our input $babaa$.

In the rest of this section, we will discuss three types of automata.

**Deterministic automata** The basic variant of an automaton operates deterministically. The automaton shown in figure 1 is an example of such a basic automaton.

Formally, we can define a deterministic finite automaton (DFA) as follows.

**Definition 2.2.1.** A DFA is a tuple $(Q, q_0, \Sigma, \delta, F)$, where

- $Q$ is a finite collection of states,
- $q_0 \in Q$ is the initial state,
- $\Sigma$ is the input alphabet,
- $\delta: Q \times \Sigma \rightarrow Q$ is the transition function,
- $F \subseteq Q$ is the collection of final states.

Before we can describe the working of a DFA, we need to extend the transition function to reflect paths instead of edges. The transitive closure of a transition function $\delta: Q \times \Sigma \rightarrow Q$ is inductively defined as

$$
\delta^* : Q \times \Sigma^* \rightarrow Q,
$$

$$
\delta^*(q, w) = \begin{cases} 
\delta^*(\delta(q, s), w') & \text{if } w = sw' \text{ with } s \in \Sigma \text{ and } w' \in \Sigma^*, \\
q & \text{otherwise}
\end{cases}
$$
On input $w \in \Sigma^*$, a DFA ends in the state $\delta^*(q_0, w)$ and accepts $w$ if and only if $\delta^*(q_0, w) \in F$. The language consisting of all words that are accepted by a DFA $A$ is denoted by $L(A)$. Given a language $L$, we say that a DFA $A$ recognizes $L$ if $L = L(A)$.

**Non-deterministic automata** A variant of a DFA allows multiple (or no) transitions from a state labeled with the same letter. The resulting automaton then operates non-deterministically. It can be thought of guessing the correct transition to take in each step, or trying out all possible transitions simultaneously.

Formally, we can define a non-deterministic finite automaton (NFA) as an extension of a DFA as follows.

**Definition 2.2.2.** An NFA is a tuple $(Q, I, \Sigma, \delta, F)$ defined exactly like a DFA, except

- $I \subseteq Q$ is the collection of initial states,
- $\delta : Q \times \Sigma \rightarrow 2^Q$ is the transition function.

Analogous to the closure of a deterministic transition function, we define the transitive closure of a non-deterministic transition function $\delta : Q \times \Sigma^* \rightarrow 2^Q$ inductively as

$$\delta^* : Q \times \Sigma^* \rightarrow 2^Q,$$

$$\delta^*(q, w) = \left\{ \begin{array}{ll}
\bigcup_{q' \in \delta(q, s)} \delta^*(q', w') & \text{if } w = sw' \text{ with } s \in \Sigma \text{ and } w' \in \Sigma^*.
\{q\} & \text{otherwise.}
\end{array} \right.$$ 

For convenience, we will overload $\delta^*$ to work for sets of states as well by

$$\delta^* : 2^Q \times \Sigma^* \rightarrow 2^Q,$$

$$\delta^*(S, w) = \bigcup_{q \in S} \delta^*(q, w).$$

On input $w \in \Sigma^*$, an NFA can reach exactly the states from $\delta^*(I, w)$ and accepts $w$ if and only if $\delta^*(I, w) \cap F \neq \emptyset$. Similar to a DFA, we define $L(A)$ to be the language of all words that are accepted by an NFA $A$, and say that an NFA $A$ recognizes a language $L$ if $L = L(A)$.

On first appearance the concept of an NFA seems to be more powerful than that of a DFA. But it turns out that any NFA can be simulated by a DFA, possibly at the expense of an exponential increase in the number states.

**Theorem 2.2.3** (Subset construction). For any NFA $A = (Q, I, \Sigma, \delta, F)$ there exists a DFA $A'$ with at most $2^{|Q|}$ states such that $L(A) = L(A')$.

**Proof.** Construct the DFA $A' = (2^Q, I, \Sigma, \delta', F')$ as follows.

- $\delta'(S, s) = \bigcup_{q \in S} \delta(q, s),$
- $F' = \{ S \in 2^Q \mid S \cap F \neq \emptyset \}.$

We will show that $\delta^*(S, w) = \delta^*(S, w)$ for all $S \in 2^Q$ and $w \in \Sigma^*$ by induction on the length of $w$. If $|w| = 0$, it clearly holds. Otherwise, we assume it holds for $w$ and prove it for $sw$ with
$s \in \Sigma$ arbitrary. We have for any $S \in 2^Q$ that

$$
\delta^*(S, sw) = \delta^*(\delta(S, s), w) = \delta^*\left(\bigcup_{q \in S} \delta(q, s), w\right)
$$

(by definition of $\delta^*$)

$$
= \bigcup_{q \in S} \bigcup_{q' \in \delta(q, s)} \delta^*(q', w)
$$

(by induction and definition of $\delta'$)

$$
= \delta^*(S, sw).
$$

(by definition of $\delta$ and $\delta^*$)

Hence, we get that $\delta^*(I, w) \in F'$ if and only if $\delta^*(I, w) \cap F \neq \emptyset$, showing that $L(A) = L(A')$. □

The advantage of using an NFA lies therefore mostly in the power of expression. An NFA is often simpler to setup and conveys the underlying idea more clearly. For example, the NFA in figure 2 recognizes the same language as the DFA in figure 1, but is smaller and easier to understand. Even more so when we generalize the example to recognize the language $L_n$ of all words over $\{a, b\}$ whose $n$th-to-last letter is $a$, for some fixed $n$.

Interestingly, this generalized example also illustrates that an exponential blow-up in the number of states when converting an NFA to a DFA cannot be avoided. Clearly, $L_n$ can be recognized by an NFA with $n + 1$ states. Now suppose that there exists a DFA $A$ with less than $2^n$ states that recognizes $L_n$. By the pigeonhole principle, there exist two distinct words $w, w' \in \{a, b\}^n$ such that $\delta^*(q_0, w) = \delta^*(q_0, w')$. Let $0 \leq j < n$ be the maximal index for which $w_j \neq w'_j$. Then we get the contradiction that $wa^j$ and $w'a^j$ are either both accepted or both rejected by $A$, but exactly one of these has an $a$ as the $n$th-to-last symbol.

**Automata with output** Another variant of a DFA allows verdicts other than accept or reject when judging an input word. Formally, we can define a *deterministic finite automaton with output* (DFAO) as follows.

**Definition 2.2.4.** A DFAO is a tuple $(Q, q_0, \Sigma, \delta, \Theta, \pi)$, where

- $Q$ is a finite collection of states,
- $q_0 \in Q$ is the initial state,
- $\Sigma$ is the input alphabet,
- $\delta: Q \times \Sigma \to Q$ is the transition function,
- $\Theta$ is the output alphabet.
• \(\pi: Q \to \Theta\) the output function.

On input \(w \in \Sigma^*\), a DFAO ends in the state \(\delta^*(q_0, w)\) and outputs \(\pi(\delta^*(q_0, w))\). Instead of inducing a language of accepted words as a DFA does, a DFAO \(A\) induces a map

\[
A: \Sigma^* \to \Theta,
A(w) = \pi(\delta^*(q_0, w)),
\]

where we use the name of the concerning DFAO as the name of the map for convenience. Given a map \(f: \Sigma^* \to \Theta\), we say that a DFAO \(A\) generates \(f\) if \(f = A\).

Let us discuss a few basic manipulations on DFAOs. The first says we can relabel the output in any way we want.

**Lemma 2.2.5.** Let \(A\) be a DFAO with output alphabet \(\Theta\) and \(f: \Theta \to \Theta'\) any map. Then the map \(f \circ A\) is generated by a DFAO.

**Proof.** Let \(A\) be given by \((Q, q_0, \Sigma, \delta, \Theta, \pi)\). Then the DFAO \((Q, q_0, \Sigma, \delta, \Theta, f \circ \pi)\) generates \(f \circ A\).

Secondly, we can fuse any two DFAOs together.

**Lemma 2.2.6.** Let \(A\) and \(A'\) be DFAOs with input alphabets \(\Sigma\) and \(\Sigma'\), and output alphabets \(\Theta\) and \(\Theta'\) respectively. Then the map \(f: (\Sigma \times \Sigma')^* \to \Theta \times \Theta',\)

\[
f((s_0, s_0') \ldots (s_{n-1}, s_{n-1}')) = (A(s_0 \ldots s_{n-1}), A'(s_0' \ldots s_{n-1}'))
\]

is generated by a DFAO.

**Proof.** Let \(A\) and \(A'\) be given by \((Q, q_0, \Sigma, \delta, \Theta, \pi)\) and \((Q', q_0', \Sigma', \delta', \Theta', \pi')\) respectively. Then the DFAO \((Q \times Q', (q_0, q_0'), \Sigma \times \Sigma', \delta'', \Theta \times \Theta', \pi'')\) given by

- \(\delta''((q, q'), (s, s')) = (\delta(q, s), \delta'(q', s'))\),
- \(\pi''((q, q')) = (\pi(q), \pi'(q'))\),

generates \(f\).

**Corollary 2.2.7.** If \(\Sigma = \Sigma'\), the map

\[
f: \Sigma^* \to \Theta \times \Theta',
f(w) = (A(w), A'(w))
\]

is also generated by a DFAO.

Lastly, we see that, loosely speaking, a DFAO can be partitioned into DFAs. It can be quite useful for e.g. unlocking the use of non-determinism when building a DFAO.

**Corollary 2.2.8.** Let \(\Sigma\) and \(\Theta\) be alphabets, and \(f: \Sigma^* \to \Theta\) a map. Then \(f\) is generated by a DFAO if and only if the language \(L_\theta = \{w \in \Sigma^* | f(w) = \theta\}\) is generated by a DFA for each \(\theta \in \Theta\).

**Proof.** \(\Rightarrow\): Let the DFAO \((Q, q_0, \Sigma, \delta, \Theta, \pi)\) generate \(f\). Then the DFA \((Q, q_0, \Sigma, \delta, F)\) with

- \(F = \{q \in Q | \pi(q) = \theta\}\)

generates \(L_\theta\).

\(\Leftarrow\): Follows from corollary 2.2.7, lemma 2.2.5, and the fact that the \(L_\theta\) partition \(\Sigma^*\).
2.3 Regular Languages

An import family of languages that has proven to be of both theoretical and practical value are the regular languages.

Definition 2.3.1. Let $\Sigma$ be an alphabet. The regular languages over $\Sigma$ are defined inductively as follows:

- the empty language $\emptyset$ is regular,
- the language $\{s\}$ is regular for every $s \in \Sigma$,
- the language $L_1 \cup L_2$ for regular languages $L_1$ and $L_2$ over $\Sigma$ is regular,
- the language $L_1 \cap L_2$ for regular languages $L_1$ and $L_2$ over $\Sigma$ is regular,
- the language $L^*$ for a regular language $L$ over $\Sigma$ is regular.

We have already seen an example of a regular language in the previous section. The language $L$ of all words over the alphabet $\{a, b\}$ that end in $aa$ or $ab$ is regular, because it equals

$$(\{a\} \cup \{b\})^* \{a\}(\{a\} \cup \{b\}).$$

That expression is a bit cluttered which makes it harder to read. Therefore, when describing regular languages via regular expressions, we often omit the curly brackets. The previous expression then becomes

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

It is no coincidence that $L$ can be described both by a regular expression and as the language accepted by an automaton. The fundamental theorem of Kleene asserts that these ways of describing a language are equivalent.

Theorem 2.3.2 (Kleene). A language $L$ is regular if and only if it is recognized by a DFA.

We often use this theorem to prove statements in the characterization that is most convenient. It is for example quite easy to prove the following lemma using automata theory, but less so using formal language theory.

Lemma 2.3.3. The following languages are regular:

- a) the language $L_1 \cap L_2$ for regular languages $L_1$ and $L_2$ over $\Sigma$ is regular,
- b) the language $L^c$ for a regular language $L$ over $\Sigma$ is regular,
- c) the language $L^R$ for a regular language $L$ over $\Sigma$ is regular.

Proof. Part [a] follows from corollary 2.2.7 and lemma 2.2.5 and part [b] from lemma 2.2.5 and part [c] let the NFA $(Q, I, \Sigma, \delta, F)$ recognize $L$. Then the NFA $(Q, F, \Sigma, \delta', I)$ that reverses transitions via

$$\delta'(q, s) = \{q' \in Q \mid q \in \delta(q', s)\}$$

recognizes $L^R$.  \[\square\]
Corollary 2.3.4. Let $\Sigma$ and $\Theta$ be alphabets, and $f : \Sigma^* \to \Theta$ a map generated by a DFAO. Then the map

$$f^R : \Sigma^* \to \Theta,$$

$$f^R(w) = f(w^R)$$

is generated by a DFAO as well.

Proof. Use corollary 2.2.8 and lemma 2.3.3c). □

On the other hand, it is almost immediate from the definition of regular languages that $L^+$ is regular if a language $L$ is, whereas that is harder to show using automata theory.

We conclude this section with a classical tool to show languages non-regular, which is not always a simple task.

**Lemma 2.3.5** (Pumping lemma). Let $L$ be a regular language over the alphabet $\Sigma$. Then there exists an integer $p$ such that every word $w \in L$ with $|w| \geq p$ can be written as $w = xyz$ with $|xy| \leq p$, $y \neq \epsilon$ and $xy^n z \in L$ for every $n \in \mathbb{N}$.

Proof. Let the DFA $A = (Q, q_0, \Sigma, \delta, F)$ recognize $L$, take $p = |Q|$, and pick $w \in L$ with $|w| \geq p$ arbitrary. Consider the sequence of at least $p + 1$ states

$$\delta^*(q_0, \epsilon), \delta^*(q_0, w_0), \ldots, \delta^*(q_0, w).$$

By the pigeonhole principle, there exist indices $0 \leq j < k \leq p$ such that

$$\delta^*(q_0, w_0 \ldots w_{j-1}) = \delta^*(q_0, w_0 \ldots w_{k-1}),$$

so we can take $x = w_0 \ldots w_{j-1}$, $y = w_j \ldots w_{k-1}$ and $z = w_{k} \ldots w_{|w|-1}$. □

Let us apply this lemma to show by way of contradiction that the language $L$ of palindromes over the alphabet $\{a, b\}$ is non-regular. For suppose it is regular, and let $p$ be the constant from the pumping lemma. Considering the word $w = a^p b a^p \in L$, we get from the pumping lemma the contradiction that $a^{p-m} b a^p \in L$ for some positive integer $m$. 
3 Numeration systems for the Gaussian integers

In this chapter we will develop several numeration systems for the Gaussian integers, which form a key component for defining automatic maps in the next chapter. Before doing so, we will give a short introduction of the Gaussian integers, followed by a general definition of numeration systems.

3.1 Gaussian integers

As introduced by Gauss in his study of biquadratic reciprocity around 1800, the Gaussian integers are the complex numbers with integer real and imaginary part, and denoted by \( \mathbb{Z}[i] \).

Under the usual addition and multiplication of complex numbers they form a commutative ring, and in fact a Euclidean domain with norm

\[
N: \mathbb{Z}[i] \to \mathbb{N}, \quad N(z) = z\overline{z} = |z|^2.
\]

The norm is multiplicative, that is, \( N(zz') = N(z)N(z') \) for any \( z, z' \in \mathbb{Z}[i] \). The units of \( \mathbb{Z}[i] \) are therefore necessarily Gaussian integers with norm 1, and these are indeed all invertible, so \( \mathbb{Z}[i]^\times = \{ \pm 1, \pm i \} \). We say \( z, z' \in \mathbb{Z}[i] \) are associates, if \( z = uz' \) for some unit \( u \).

Since \( \mathbb{Z}[i] \) is a Euclidean domain, it is also a unique factorization domain (UFD). This means that any non-zero Gaussian integer can be written as a product of a unit and Gaussian primes, where the product is uniquely determined up to the order of its factors, and the primes are uniquely determined up to multiplication by a unit. A Gaussian prime is a non-zero non-unit element \( \pi \in \mathbb{Z}[i] \) with the property that \( \pi \mid zz' \) for some \( z, z' \in \mathbb{Z}[i] \) implies that \( \pi \mid z \) or \( \pi \mid z' \).

In a UFD, a prime element is irreducible, and vice versa.

What are the prime elements of \( \mathbb{Z}[i] \)? Surely, if \( \pi \) is a Gaussian prime, then it divides \( \pi \overline{\pi} = N(\pi) \). By the fundamental theorem of arithmetic, we can write \( N(\pi) \) as a product of natural primes (primes in \( \mathbb{N} \)), and by definition, \( \pi \) divides some natural prime \( p \) in that product.

Hence, we only need to inspect the factorization of natural primes in order to locate all Gaussian primes.

So let \( p \in \mathbb{N} \) be a natural prime, and suppose it has a non-trivial factorization \( p = zz' \) for some \( z, z' \in \mathbb{Z}[i] \). Then necessarily \( N(z) = N(z') = p \), and hence, both \( z \) and \( z' \) are irreducible and therefore prime. Now from Fermat’s theorem on sums of two squares we get that \( p \) can be written as \( x^2 + y^2 \) for some \( x, y \in \mathbb{N} \) if and only if \( p \not\equiv 3 \mod 4 \). Thus, in case \( p \equiv 3 \mod 4 \), no such \( z \) or \( z' \) can exist, proving that \( p \) is irreducible and a Gaussian prime.

Otherwise, \( p \) factors non-trivially as the prime product \( (x + yi)(x - yi) \).

The only question remaining is, can the primes \( x + yi \) and \( x - yi \) with norm \( p \not\equiv 3 \mod 4 \) be associates? We have that \( x - yi \mid x + yi \) if and only if \( p \mid (x + yi)^2 = p - 2y^2 + 2xyi \) if and only if \( p \mid 2y^2 + 2x \), because \( p \) does not divide \( x \) or \( y \) since \( 0 < |x|, |y| < p \).

In summary, the Gaussian primes are the associates of \( 1 + i, x + yi, x - yi \) and \( p \), for all natural primes \( x^2 + y^2 \equiv 1 \mod 4 \) and \( p \equiv 3 \mod 4 \).

The following easy lemma will prove useful in the last chapter.

Lemma 3.1.1. Let \( z \in \mathbb{Z}[i] \) be given. Then \( z^n \in \mathbb{Z} \) for some positive \( n \in \mathbb{N} \) if and only if \( z \) is associated to \( (1 + i)^km \) for some \( k, m \in \mathbb{N} \).

Proof. The backward direction is clear, and the forward direction follows from

\[
\pi \mid z \iff \pi^n \mid z^n \iff \overline{\pi}^n \mid z^n \iff \pi \mid z
\]

\(^2\)Field norm, not to be confused with vector norm.
for a Gaussian prime $\pi$, where $n > 0$ is such that $z^n \in \mathbb{Z}$.

With the structure of $\mathbb{Z}[i]$ laid out, let us take a look at an operation we will often perform when constructing digit sets. Taking the quotient of $\mathbb{Z}[i]$ with $\beta \mathbb{Z}[i]$, which we will abbreviate as $\mathbb{Z}[i]/\beta$, for some non-zero $\beta \in \mathbb{Z}[i]$. The Gaussian integer multiples of $\beta$ form a lattice in the complex plane. A fundamental domain of such a lattice is a subset of $\mathbb{C}$ that contains exactly one residue of each residue class of $\mathbb{C}/\beta \mathbb{Z}[i]$. The standard fundamental domain of $\beta \mathbb{Z}[i]$ in $\mathbb{C}$ is given in the following lemma. See also figure 3.

Lemma 3.1.2. Let $\beta \in \mathbb{Z}[i]$ be non-zero. Then $F = [0,1)\beta + [0,1)i\beta$ is a fundamental domain of the lattice $\beta \mathbb{Z}[i]$ in $\mathbb{C}$.

Proof. First, we show that $F$ contains at least one residue of each residue class of $\mathbb{C}/\beta \mathbb{Z}[i]$. Pick $z \in \mathbb{C}$ arbitrary, and let $z' = [\Re(z)] + [\Im(z)]i$. Then $z - z' \beta \in F$, because

$$\Re(z_{\beta} - z') = \Re(z_{\beta}) - [\Re(z_{\beta})] \in [0,1)$$

and likewise for the imaginary part.

Second, we show that $F$ contains at most one residue of each residue class of $\mathbb{C}/\beta \mathbb{Z}[i]$. Suppose that $z, z' \in F$ are two residues of the same class, so $z - z' \in \beta \mathbb{Z}[i]$. Since $z - z' \in F - F = (-1,1)\beta + (-1,1)i\beta$, we must have that $z - z' = 0$.

Let us determine the cardinality of $\mathbb{Z}[i]/\beta$, that is, the number of Gaussian integers in a fundamental domain of $\beta \mathbb{Z}[i]$ in $\mathbb{C}$. One possible approach is depicted in figure 4, where we
Figure 4: Two fundamental domains of the lattice $\beta\mathbb{Z}[i]$ in $\mathbb{C}$ for $\beta = 3 + i$. The equivalence can be shown by carefully rearranging the domains.

take the standard fundamental domain of the previous lemma, and very carefully transform it to a fundamental domain geometrically consisting of two axis-aligned squares with sides $|\Re(\beta)|$ and $|\Im(\beta)|$ respectively, showing that $|\mathbb{Z}[i]/\beta| = N(\beta)$. It is however a bit tedious to make this precise, so instead we will resort to Pick’s theorem from [Pic99] that can also help out when dealing with more complicated fundamental domains.

**Theorem 3.1.3** (Pick). Let $P$ be a simple polygon in $\mathbb{R}^2$ with integer points as vertices. Then the area $A$ of $P$ satisfies

$$A = I + B/2 - 1$$

where $I$ is the number of integer points in the interior of $P$, and $B$ is the number of integer points on the boundary of $P$.

Applying this theorem to the square with vertices 0, $\beta$, $(1 + i)\beta$ and $i\beta$ that contains the standard fundamental domain $F = ((0,1)\beta + (0,1)i\beta) \cup (0,1)\beta \cup (0,1)i\beta \cup \{0\}$ of $\beta\mathbb{Z}[i]$ in $\mathbb{C}$, we get that

$$N(\beta) = A = I + (B - 4)/2 + 1 = |F \cap \mathbb{Z}[i]| = |\mathbb{Z}[i]/\beta|.$$

Yet another way of finding $|\mathbb{Z}[i]/\beta|$ is due to Gauss, and basically yields an axis-aligned rectangle as a fundamental domain. It can be proven for example using an elementary result in number theory called Bézout’s lemma.

**Lemma 3.1.4** (Bézout). Let $a$, $b$ and $c$ be integers with $a$ and $b$ not both zero. Then there exist integers $x$ and $y$ such that $ax + by = c$ if and only if $\gcd(a, b) \mid c$. Moreover, if the pair $(x, y)$ is a solution, all solution pairs are given by $(x + n\frac{b}{\gcd(a, b)}, \ y - n\frac{a}{\gcd(a, b)})$ for $n \in \mathbb{Z}$, where $g = \gcd(a, b)$.

**Lemma 3.1.5** (Gauss). Let $\beta = a + bi \in \mathbb{Z}[i]$ be non-zero, $g = |\gcd(a, b)|$, and $N = N(\beta)/g$. Then $F = [0, 1)N + [0, 1)i$ is a fundamental domain of the lattice $\beta\mathbb{Z}[i]$ in $\mathbb{C}$.

**Proof.** First, we show that $F$ contains at least one residue of each residue class of $\mathbb{C}/\beta\mathbb{Z}[i]$. Pick $z \in \mathbb{C}$ arbitrary, and use Bézout’s lemma to find integers $x$ and $y$ such that $ax + by = \lfloor 3(\frac{z}{\beta}) \rfloor g$. 

Let $z' = y + xi \in \mathbb{Z}[i]$ and $z'' = \lfloor \Re(\frac{z'}{N}) \rfloor g \in \mathbb{Z}[i]$. Then we get that $z - z'\beta - z''\beta$ has real part

$$
\Re(z - z'\beta - z''\beta) = \Re(z - z'\beta) - \lfloor \Re(\frac{z'}{N}) \rfloor N
$$

and imaginary part

$$
\Im(z - z'\beta - z''\beta) = \Im(z) - (ax + by)
$$

for some integers $x$ and $y$. Since $z - z' \in F - F = (-1, 1)N + (-1, 1)ig$, we have that

$$
|\Re(z - z')| < N \quad \text{and} \quad |\Im(z - z')| < g.
$$

Therefore, we know from Bézout’s lemma that $\Im(z - z') = ax + by = 0$, and that $x = -n\frac{b}{g}$ and $y = n\frac{a}{g}$ for some integer $n$. Then

$$
|\Re(z - z')| = |ay - bx| = |n| N < N,
$$

so $z - z' = 0$.

\section{3.2 Numeration systems}

In this section we will define general numeration systems for commutative semirings, and introduce related terminology.

Common requirements on numeration systems are demanding zero to be a digit, and using no more digits than absolutely necessary. Here, we will allow any set of digits to be used, perhaps more than really necessary, but ask to supply a dictionary to tell which digit sequences are valid. Note that it is implicit in the following definition that the number of digits is finite.

\begin{definition}
A \textit{numeration system} for a commutative semiring $S$ is a tuple $(B, D, R)$, where

- $B: \mathbb{N} \to S$ is the base sequence,
- $D \subseteq S$ is an alphabet, the digit set,
- $R \subseteq D^*$ is a language, the representation set.

Given a numeration system $\mathcal{N} = (B, D, R)$ for a commutative semiring $(S, +, \cdot)$ we evaluate a word $w \in D^*$ of length $n$ in the usual way by

$$
eval_{\mathcal{N}}: D^* \to S,$$

$$
eval_{\mathcal{N}}(w) = w_0 \cdot B_{n-1} + w_1 \cdot B_{n-2} + \ldots + w_{n-1} \cdot B_0.
$$

The ordinary decimal system for $\mathbb{N}$ is for example given by $\mathcal{N}' = (B', D', R')$ with $B'_n = 10^n$, $D' = \{0, 1, \ldots, 9\}$ and $R' = D'^* \setminus (0^+ D'^*)$, and evaluating the decimal word $189 \in R'$ simply comes down to

$$
eval_{\mathcal{N}'}(189) = 1 \cdot 10^2 + 8 \cdot 10 + 9 \cdot 1.$$

16
Here, the order of digits from most to least significant is somewhat arbitrary, just like automata and the order of their input. We could also have opted to reverse the order of digits and adjust other definitions accordingly, it works just as well.

Before we move on to the rest of the definitions, let us give one additional example.

**Example 3.2.2.** A more exotic example of a numeration system for \( \mathbb{N} \) with an interesting base sequence and representation set is due to \([\text{Zec72}]\). Zeckendorf’s theorem says that every positive integer can be written uniquely as a sum of positive, strictly increasing, non-consecutive Fibonacci numbers. We can turn this theorem into a numeration system \( \mathcal{N}'' = (B'', D'', R'') \) for \( \mathbb{N} \) by using \( B''_n = F_{n+2}, \ D'' = \{0, 1\} \) and \( R'' = \epsilon \cup 1(0 \cup 01)^* \), where \( (F_n)_{n \in \mathbb{N}} \) is the Fibonacci sequence with \( F_0 = 0 \) and \( F_1 = 1 \). If we enumerate the words in \( R'' \) in order of their evaluation, we get

\[
R'' = \{ \epsilon, 1, 10, 100, 101, 1000, 1010, 10000, 10001, \ldots \}.
\]

A word \( w \in D^* \) is an \( \mathcal{N} \)-expansion of \( s \in S \) if \( \text{eval}_\mathcal{N}(w) = s \). It is an \( \mathcal{N} \)-representation of \( s \) if in addition \( w \in R \). When the numeration system \( \mathcal{N} \) is clear from the context, we may drop it from both terms.

We say a numeration system is **injective** (surjective, bijective\footnote{Not to be confused with the particular numeration systems \( (n \mapsto b^n, D, D^*) \) with \( D = \{1, 2, \ldots, b\} \) for \( \mathbb{N} \) that are known as the bijective numeration systems.}) if the restriction of its evaluation map to \( R \) is. Put otherwise, a numeration system is injective (surjective, bijective) if every element of \( S \) has at most (at least, exactly) one representation. In a bijective system, the representation of elements from \( S \) can be retrieved by

\[
\text{repr}_\mathcal{N} : S \rightarrow R, \\
\text{repr}_\mathcal{N} = (\text{eval}_\mathcal{N}|_R)^{-1}.
\]

A numeration system is **regular**, if \( R \) is regular. A numeration system is **economical**, if for every \( s \in S \) each of its representations is at least as short as any of its expansions. That is, every representation is as short as possible.

A **radix system** is a numeration system where the base sequence \( B \) is given by \( B_n = b^n \) for some **base** (or **radix**) \( 0 \neq b \in S \).\footnote{A zero base is seldom of interest, so we choose to exclude it by definition.} Such systems are usually denoted by a tuple \( (b, D, R) \) with the base sequence being implied by \( b \). A complete residue system \( (\text{CRS}) \) for \( S \) is a radix system with base \( b \in S \) where the digit set contains exactly one residue of each residue class in \( S/b \). Here we mean with \( S/I \) for an ideal \( I \) of \( S \) the quotient \( S/\sim \) where \( s \sim t \) if and only if \( s + i = t + j \) for some \( i, j \in I \). A radix system is basic if it is an economical, bijective CRS.

Generalizing the concept of leading zeros, a non-empty prefix \( v \) of a word \( vw \in D^* \) satisfying

\[
\text{eval}_\mathcal{N}(v) b^{|w|} = 0
\]
is called a zero lead. In economical radix systems, representations are necessarily words without zero leads. The converse is true as well when we impart some additional structure to both the numeration system and semiring.

We say a semiring \( S \) is cancellative under multiplication by \( b \), if for all \( s, t \in S \) we have that \( sb = tb \implies s = t \). If \( S \) is a ring, it is equivalent to say that \( b \) is no zero divisor.

**Lemma 3.2.3.** Let \( \mathcal{N} = (b, D, R) \) be a CRS for the commutative ring \( S \), where \( b \) is no zero divisor. If \( w \in D^* \) is an expansion of \( s \in S \), then
\[ w \text{ is the unique expansion of } s \text{ of length } |w|, \]
\[ w \text{ contains every shorter expansion of } s \text{ as a suffix,} \]
\[ w \text{ has no zero lead if and only if } w \text{ is the shortest expansion of } s. \]

In particular, if \( N \) is economical, it is injective.

Proof. We can write \( s \) uniquely as \( s = tb + d \) with \( t \in S \) and \( d \in D \). Namely, \( d \) is uniquely determined by \( s \mod b \) since \( N \) is a CRS, and \( t \) is uniquely determined since the factor \( b \) can be canceled. Continuing writing \( t \) out and so on, we see that we can write \( s \) for every \( n \in N \) uniquely as \( s = s_n b^n + \text{eval}_N(w) \) with \( s_n \in S \) and \( w \in D^n \). Of course, \( w \) is an expansion of \( s \) of length \( n \) if and only if \( s_n = 0 \), and the result follows.

Corollary 3.2.4. Let \( N \) in addition be basic. If \( 0 \) has a (unique) shortest non-empty expansion \( w \in D^* \), then \( R = D^* \setminus (w^+ D^*) \), and \( R = D^* \) otherwise. In particular, \( N \) is regular.

Proof. Pick an arbitrary expansion \( v \in D^* \). If \( v \) has a zero lead, it is not a representation because \( N \) is economical, so assume it has not. From the previous lemma we then get that \( v \) is the unique shortest expansion of \( s = \text{eval}_N(v) \), so it represents \( s \) as \( N \) is surjective and economical.

Now any zero lead is an expansion of \( 0 \) since the factor \( b \) can be canceled. Hence, if \( 0 \) has no shortest non-empty expansion \( w \in D^+ \), no zero leads exist and \( R = D^* \). Otherwise, we get from the previous lemma that \( w \) is unique and that \( w^* \) forms precisely all expansions of \( 0 \), so \( R = D^* \setminus (w^+ D^*) \).

3.3 Numeration systems for the Gaussian integers

In this section we will discuss several well-known radix systems for \( \mathbb{Z}[i] \) in every base that are both bijective and regular, and finish with a small technical bit.

We will only consider representing \( \mathbb{Z}[i] \) using a single system. This means that we do not treat real and imaginary parts separately, although this can effectively be achieved in certain bases. For example, the regular system \( N_{NW} = (2, D_{NW}, R_{NW}) \) for \( \mathbb{Z}[i] \) with
\[
D_{NW} = \{0, 1\} + \{0, 1\}i,
R_{NW} = D_{NW}^* \setminus (0^+ D_{NW}^*),
\]
can be viewed to represent all Gaussian integers in the north-west quadrant by expressing their real and imaginary parts in binary.

We also do not consider extending our numeration systems by adding sign symbols. The same result can be had by including every sign-digit combination in our digit set and choosing our representations appropriately. So we can for instance get a surjective regular radix system (that can easily be made bijective as well) in base 2 for \( \mathbb{Z}[i] \) by combining a variant of the previous example for each quadrant:
\[(2, D_{NW} \cup D_{SW} \cup D_{SE} \cup D_{NE}, R_{NW} \cup R_{SW} \cup R_{SE} \cup R_{NE}).\]
Figure 5: Dragon curve of order 7: the curve that arises by folding a strip of paper 7 times down the middle, followed by unfolding the paper strip with 90 degree angles at the creases.

**Canonical numeration systems**  Canonical numeration systems such as our binary or decimal systems for $\mathbb{N}$, are in a sense quite 'natural'. A radix system $\mathcal{N} = (\beta, D, R)$ for $\mathbb{Z}[i]$ is canonical if the digit set $D = \{0, 1, \ldots, N(\beta) - 1\}$. For which bases are such systems bijective? Obviously, a necessary condition is that $\mathcal{N}$ is a CRS because $|D| = N(\beta)$. That means that $g = \gcd(\Re(\beta), \Im(\beta)) = 1$, since zero is congruent to $(\beta/g)\beta = N(\beta)/g$ modulo $\beta$. That is not the only condition however, and a complete characterization is given in [KS75].

**Theorem 3.3.1** ([KS75]). There exists a canonical basic numeration system $(\beta, D, R)$ for $\mathbb{Z}[i]$ if and only if $\beta = -b \pm i$ for some positive integer $b$.

**Revolving numeration systems**  An example of a more exotic radix system $\mathcal{N} = (\beta, D, R)$ is the revolving system for base $\beta = 1 + i$ (and of course $1 - i$ by conjugation) from [DK70]. In that paper it is shown that the revolving system is intimately connected to the classical Dragon curve from figure 5. Let us describe the system in detail.

The digit set $D = \{0, 1, -i, -1, i\}$, so it is not a CRS in our sense since the non-zero digits are members of the same residue class of $\mathbb{Z}[i]/\beta$. However, one may view it as a CRS with a digit set that changes during expansion. Namely, representations are bound to the rule that its non-zero digits from most to least significant follow the cycle

$$
\begin{align*}
\circlearrowleft & \ i \\
\circlearrowright & \ -1 \quad 1 \\
\circlearrowleft & \ -i
\end{align*}
$$

which is where the name ‘revolving’ derives from. For instance, $-2 + i$ can simply be represented as $(1, 0, 0, -i)$, but also as $(1, -i, -1, i, 1)$. The representation set $R$ can be described as the language accepted by the NFA from figure 6. In particular, this means that $\mathcal{N}$ is regular. How about injectivity and surjectivity? The previous example already shows that $\mathcal{N}$ is not injective, but the following theorem shows that we can derive a regular bijective revolving system by replacing $R$ with $(R \cap D^*d0^*) \cup \{\epsilon\}$ with $d \in D$ a unit of choice.

**Theorem 3.3.2** ([DK70]). Every non-zero Gaussian integer has exactly four revolving representations, one each in which the right-most non-zero digit takes on the values 1, $i$, $-1$, $-i$. 
Figure 6: NFA that recognizes revolving representations. The state names indicate which non-zero digit is valid.

On the other hand, taking only the revolving expansions that start with the digit 1 as representations does not result in a bijective revolving system, as it is shown in [DK70] that these expansions are precisely those lattice points visited by the dragon curve that has (0, 1) as its initial segment.

The previous theorem can be used to derive what is perhaps the most interesting property of the dragon curve. The curve never intersects itself. Moreover, four copies rotated by respectively 0, 90, 180 and 270 degrees never intersect with each other or themselves, and fit perfectly. Together, they traverse every line segment of length 1 in the lattice \( \mathbb{Z}[i] \) exactly once. See figure 7 for a small example of the neat intertwining after the first few folds.

Furthermore, it is remarked in [DK70] that the four revolving expansions of a non-zero Gaussian integer can be grouped in two pairs of ‘similar’ representations. This follows from the identity

\[ u = u \beta - iu \]  

for any \( u \in \mathbb{Z}[i] \). Namely, let \( u \in D \) be the least significant non-zero digit of a given revolving representation \( w \in R \). If the digit \( d \in D \) immediately preceding \( u \) in \( w \) is 0 (or non-existent), we may safely replace the subword \( du \) (or \( u \)) by the word \( (u, -iu) \) in \( w \) without affecting its evaluation or its revolving property. Otherwise, we may safely replace it by the word 0d in \( w \).

Finally, the intimate connection between the dragon curve and the revolving system is made apparent in [DK70] with an alternative generation of the dragon curve. Given \( w \in R \) with \( \text{eval}_X(w) \neq 0 \), alternate the following two steps indefinitely to get a sequence of revolving representations (starting with \( w \)) that describe consecutive points of a dragon curve upon evaluation.

1. (Turn) Replace \( w \) by its similar representation using (1).
2. (Move) Replace the last digit \( d \in D \) of \( w \) by 0 if \( d \neq 0 \), and by the unique non-zero \( u \in D \) such that the result remains revolving otherwise.

See for example the first few steps in figure 8 starting with \( w = 1 \).
Figure 7: Four rotated copies of the dragon curve eventually traverse every line segment of length 1 in the lattice $\mathbb{Z}[i]$ exactly once.

Figure 8: First few steps in generating the dragon curve by alternately Turning and Moving revolving representations starting from 1. Here, the digit $-1$ is denoted concisely as $\bar{1}$. 
Basic numeration systems  The two types of bijective numeration systems we just discussed cover only a limited selection of bases. We would of course like to expand that selection to include all possible bases, or at least all bases with sufficiently big norms. In the literature (e.g. [AS03]) it is mentioned that among other things the following result of interest can be found in [DDG78]:

**Theorem 3.3.3 ([DDG78]).** For every base with norm at least 5, there exists a bijective radix system for $\mathbb{Z}[i]$.

Unfortunately, that report proved hard to find, so we reproduce the result here. Specifically, we show that every base admits a basic numeration system, apart from the bases $1 \pm i$, 2 or a unit. We start by showing that we only need to worry about the representability of Gaussian integers with small norms.

Note that $|\text{eval}_N(d)|$ for a digit $d \in D$ is an unfortunate quirky way of taking the modulus of the letter $d$.

**Lemma 3.3.4.** Let $\mathcal{N} = (\beta, D, R)$ be a CRS for $\mathbb{Z}[i]$ with $|\beta| > 1$ such that every $z \in \mathbb{Z}[i]$ with $|z| < c\Delta$ has a representation, for some real $c > \frac{1}{|\beta|-1}$ and $\Delta = \max_{d \in D} |\text{eval}_N(d)|$. Then $\mathcal{N}$ can be made to be bijective.

**Proof.** By lemma 3.2.3 it is enough to show that every $z \in \mathbb{Z}[i]$ has an expansion, and we will do so by induction on $N(z)$. If $N(z) < c^2\Delta^2$, this holds by assumption. Otherwise, write $z = z'\beta + d$ uniquely with $z' \in \mathbb{Z}[i]$ and $d \in D$. Then $z'$ has an expansion $w \in D^*$ by induction, because

$$|z - d| \leq |z| + |\text{eval}_N(d)| \leq |z| + \Delta \leq \left(1 + \frac{1}{c}\right)|z| < |\beta| |z|$$

implies that $N(z') < N(z)$. Hence, $z$ has expansion $wd$. \hfill $\Box$

The previous lemma suggests that we choose our digits as close to zero as possible, lest we want to have more work on our hands. Of course, we cannot blindly take only the Gaussian integers with smallest norms as our digits, as can be seen in figure 9. Instead, we can ensure the following upperbound on the size of our digits.

**Lemma 3.3.5.** Let $\beta \in \mathbb{Z}[i]$ be non-zero, and $D = \{z \in \mathbb{Z}[i] \mid |z| \leq \frac{1}{2}\sqrt{2}|\beta|\}$. Then $D$ contains at least one residue of each residue class in $\mathbb{Z}[i]/\beta$. 22
Proof. The centered fundamental domain \([-\frac{1}{2}, \frac{1}{2})\beta + [-\frac{1}{2}, \frac{1}{2})i\beta\) contains exactly one residue of each residue class of \(\mathbb{Z}[i]/\beta\), and each such residue is part of \(D\).

Note that the bound from the previous lemma is tight for bases with even norms. For bases with odd norms, it turns out there is a little room for improvement, but the gain is negligible and unnecessary for large bases.

Another aspect that helps satisfying the condition of lemma 3.3.4 is including as many Gaussian integers with small norms as possible in our digit set.

Lemma 3.3.6. Let \(\beta \in \mathbb{Z}[i]\), and \(D = \{z \in \mathbb{Z}[i] \mid |z| < \frac{1}{2} \beta\}\). Then \(D\) contains at most one residue of each residue class in \(\mathbb{Z}[i]/\beta\).

Proof. We may assume that \(\beta \neq 0\). Suppose \(d, d' \in D\) are such that \(d - d' = z\beta\) for some \(z \in \mathbb{Z}[i]\). Then

\[N(z)N(\beta) = N(d - d') \leq (|d| + |d'|)^2 < N(\beta),\]

so \(N(z) = 0\) and therefore \(d = d'\).

The previous lemmas already give us most of the result we are after.

Theorem 3.3.7. Let \(\beta \in \mathbb{Z}[i]\) be a base not equal to \(1 \pm i, 2\) or a unit. Then there exists a basic numeration system \((\beta, D, R)\). In particular, we can take \(D = \{[-\frac{1}{2}, \frac{1}{2})\beta + [-\frac{1}{2}, \frac{1}{2})i\beta\} \cap \mathbb{Z}[i]\).

Proof. Let \(D\) be as stated. We will show that \(N = (\beta, D, R)\) satisfies lemma 3.3.4 for appropriate \(R\) and \(c\) in several cases.

- \(N(\beta) \geq 6\): Since \(\Delta \leq \frac{1}{2} \sqrt{2} |\beta|\), we can take \(c = \frac{1}{2} \sqrt{2}\), and \(R = D\).
- \(N(\beta) = 5\): \(D\) simplifies to \(\{0, \pm 1, \pm i\}\), so \(\Delta = 1\) and we can take \(c = 1\) and \(R = \{\epsilon\}\).
- \(\beta = -2\): \(D\) simplifies to \(\{0, 1, i, 1+i\}\), so \(\Delta = \sqrt{2}\) and we can take \(c = \Delta\) and

\[R = D \cup \{(1, 1+i), (1, 1), (1+i, 1+i), (i, i), (i, 1+i)\}.

- \(\beta = \pm 2i\): It suffices to consider only \(\beta = 2i\), the other follows by conjugation. \(D\) simplifies to \(\{0, 1, -i, 1-i\}\), so \(\Delta = \sqrt{2}\) and we can take \(c = \Delta\) and

\[R = D \cup \{(1, 1-i), (1, -i), (1, 1+i, 1+i), (1, -i, 1), (1, -i, 1-i)\}.

- \(\beta = -1 \pm i\): It suffices to consider only \(\beta = -1 + i\), the other follows by conjugation. \(D\) simplifies to \(\{0, 1\}\), so \(\Delta = 1\) and we can take \(c = \sqrt{6}\). That leaves twenty-one \(z \in \mathbb{Z}[i]\) with \(N(z) \leq 5\) to represent, but fortunately we can get that number down to the following five:

\[i = \text{eval}_N(11), \quad -2 + i = \text{eval}_N(11111),\]
\[-i = \text{eval}_N(111), \quad -2 - i = \text{eval}_N(1110101),\]
\[-1 = \text{eval}_N(11101).\]

Any other \(z \in \mathbb{Z}[i]\) can be written uniquely as \(z = z'\beta + d\) with \(z' \in \mathbb{Z}[i]\) and \(d \in D\), and satisfies \(N(z') < N(z)\). Namely, if \(z\) has even norm then \(N(z') = N(z)/2 < N(z)\), and otherwise

\[N(z') = N(z - 1)/2 < N(z) \iff N(z) - 2|\Re(z)| + 1 < 2N(z) \iff 2 < N(z + 1) \iff z + 1 \notin \{0, \pm 1 \pm i\}.

Hence, \(z\) can inductively be given the representation of \(z'\) followed by the digit \(d\).
A few remarks on the previous theorem are in order. First, for bases with norm at least 6, we can see that any digit set satisfying both lemma 3.3.5 and 3.3.6 can be used in place of the chosen digit set from the centered fundamental domain.

Second, note that the last part of the proof is effectively a direct proof that does not use lemma 3.3.4 and is subsumed by theorem 3.3.1. In that case, we can profit from the small digit set to calculate the norm of $z'$ exactly, instead of upperbounding it as in lemma 3.3.4, cutting down the number of Gaussian integers to represent explicitly considerably.

Finally, why did we have to exclude a few small cases? Their impossibility can be explained as follows.

Lemma 3.3.8. No surjective CRS exists for $\mathbb{Z}[i]$ with base $1 \pm i$, 2 or a unit.

Proof. Suppose on the contrary that $\mathcal{N} = (\beta, D, R)$ is a surjective CRS with $\beta = 1 - u$ and $u \neq 1$ a unit, and let $d \in D$ be non-zero. Then $u^{-1}d \neq 0$, so it is not represented by $\epsilon$. Also, $u^{-1}d \neq d$ and $u^{-1}d \equiv d \mod \beta$, so it is not represented by a single digit since $\mathcal{N}$ is a CRS. But we can write $u^{-1}d$ uniquely as $u^{-1}d = u^{-1}d\beta + d$, so it has no (terminating) expansion in contradiction with the surjective property.

Unary numeration systems. A radix system is unary if its base is a unit. Even though we have seen in lemma 3.3.8 that no CRS exists for $\mathbb{Z}[i]$ in a unary base, does not mean we cannot construct a regular bijective unary radix system. There are in fact many possible choices, one of which is the system $\mathcal{N} = (1, D, R)$ with

$$D = \{-1, 1, -\iota, \iota\},$$

$$R = \left( (\bar{\iota})^* \cup 1^* \right) \left( (-\bar{\iota})^* \cup \bar{\iota}^* \right).$$

If we want such a unary system in another base, we can simply adjust $\mathcal{N}$ by adding the digit 0 and periodically interleaving representations with 0 whenever the corresponding base power is not equal to 1. For instance, $\mathcal{N} = (-1, D, R)$ with

$$D = \{0, -1, 1, -\iota, \iota\},$$

$$R = \left( (-1, 0)^* \cup (10)^* \right) \left( (-\bar{\iota}, 0)^* \cup (\bar{\iota}0)^* \right),$$

is a bijective regular unary radix system. However, it turns out that we have little use for unary systems so we will leave it at this.

Balanced numeration systems. Analogous to the well-known balanced ternary system $(3, D, R)$ for $\mathbb{Z}$ with

$$D = \{-1, 0, 1\},$$

$$R = D^* \setminus (0^* D^*),$$

we can construct balanced radix systems $(\beta, D, R)$ for $\mathbb{Z}[i]$ for bases with odd norms. Here, we understand such a radix system to be balanced, if its digit set is invariant under multiplication by a unit. Advantages of such systems include that it is very easy to derive expansions for
Lemma 3.3.9. Let $\beta \in \mathbb{Z}[i]$ be a base with odd norm. Then the fundamental domain $[-\frac{1}{2}, \frac{1}{2})\beta + [-\frac{1}{2}, \frac{1}{2})i\beta$ has no Gaussian integer on its boundary.

Proof. By symmetry it suffices to show that $(q - \frac{1}{2})i\beta \notin \mathbb{Z}[i]$ for each rational $q$. Write $\beta = a + bi$, and suppose on the contrary that $(q - i)\beta = (qa + b) + (qb - a)i \in 2\mathbb{Z}[i]$ for some rational $q$. Because $N(\beta)$ is odd, $a$ is even if and only if $b$ is odd. Assume that $a$ is even, the other case is similar. Then $qa$ has to be an odd integer, so the denominator of $q$ must be even in reduced form. But then we get the contradiction that $qb - a$ is non-integer. \qed

Following the remark after theorem 3.3.7, we have that any balanced digit set that satisfies both lemma 3.3.5 and 3.3.6 leads to a balanced basic numeration system for a base with odd norm at least 9. We can build such digit sets incrementally, starting with the digit set balanced at each step. So the only thing left to check is that none of the associated residues are valid options for inclusion as well, because we keep the digit set balanced at each step. So the only thing left to check is that none of the associated residues share a residue class modulo $\beta$.

Lemma 3.3.10. Let $\beta \in \mathbb{Z}[i]$ be a base with odd norm, $d \in \mathbb{Z}[i]$ with positive norm less than $N(\beta)$ and $u \in \mathbb{Z}[i]$ a unit. Then $d \equiv ud \mod \beta$ if and only if $u = 1$.

Proof. The backward direction is trivial, so let us show the forward direction by means of contradiction. Suppose $(1-u)d = z\beta$ for some $z \in \mathbb{Z}[i]$ and $u \neq 1$. Since $N(\beta)$ is odd, $1+i$ does not divide $\beta$. Hence, $1-u$ has to divide $d$, making $d$ a multiple of $\beta$. But that is impossible as $0 < N(d) < N(\beta)$! \qed

Compact numeration systems Compact numeration systems are merely technical in nature. We say a radix system $\mathcal{N} = (\beta, D, R)$ for $\mathbb{Z}[i]$ is compact, if $\text{eval}_\mathcal{N}(d) < |\beta| - 1$ for each $d \in D$. From theorem 3.3.7 we can see that there exists a compact basic numeration system for every base with modulus at least $2 + \sqrt{2}$, or norm at least 12, and that is enough for our purposes. The reason we are interested in this type of system is because it allows the following estimation of the length of a representation of $z \in \mathbb{Z}[i]$ based on its modulus.

Lemma 3.3.11. Let $\mathcal{N} = (\beta, D, R)$ be a compact basic numeration system for $\mathbb{Z}[i]$. Then there exists a real $\epsilon > 0$ such that for all $z \in \mathbb{Z}[i]$ and $n \in \mathbb{N}$ we have that

$$|z| < \epsilon |\beta|^n \implies |\text{repr}_\mathcal{N}(z)| \leq n.$$
$w$ has no zero lead and equals $\text{repr}_N(z')$ for some $z' \in \mathbb{Z}[i]$ by lemma 3.2.4 and $z'$ satisfies

$$|z'| = \left| \frac{z - d}{\beta} \right| \leq \frac{|z|}{|\beta|} + \frac{\Delta}{|\beta|}$$

$$\leq \epsilon |\beta|^{-1} + \frac{\Delta}{|\beta|} \frac{1}{|\beta| - 1} + \frac{\Delta}{|\beta|}$$

$$= \epsilon |\beta|^{-1} + \frac{\Delta}{|\beta| - 1}.$$ 

Therefore, we get by induction that $|\text{repr}_N(z)| = |\text{repr}_N(z')| + 1 \leq n$. 

Of course, we can do an easy converse estimate as well.

**Lemma 3.3.12.** Let $N = (\beta, D, R)$ be a compact basic system for $\mathbb{Z}[i]$. Then for all $z \in \mathbb{Z}[i]$ and $n \in \mathbb{N}$ we have that

$$|\text{repr}_N(z)| \leq n \implies |z| < |\beta|^n - 1.$$ 

**Proof.** For $w = \text{repr}_N(z)$ we have that

$$|z| = |\text{eval}_N(w)|$$

$$\leq |\text{eval}_N(w_0)||\beta|^{n-1} + |\text{eval}_N(w_1)||\beta|^{n-2} + \ldots + |\text{eval}_N(w_{n-1})|$$

$$< (|\beta| - 1)|\beta|^{n-1} + (|\beta| - 1)|\beta|^{n-2} + \ldots + |\beta| - 1$$

$$= |\beta|^n - 1.\qed$$
4 Automatic maps on the Gaussian integers

In the previous chapters we discussed the key ingredients for defining automatic maps, viz. automata and numeration systems. We will use these here to formalize automatic maps with respect to a given base, and finish by comparing the classes of automatic maps in different bases.

4.1 General definition and basic properties

In this section we investigate how we can generalize the concept of automatic sequences that was introduced by Cobham around 1970 to automatic maps on the Gaussian integers. Since it is not much more difficult to work with arbitrary commutative rings rather than \( \mathbb{Z}[i] \) in specific, we will do that instead. The majority of the results given here can be found in [ACG+97] where the very same general question is studied for semirings, but the material will differ in some details and the way it is presented.

In section 2.2 we discussed finite automata and the maps they induce. Here, we are particularly interested in the case where the input words for automata with output are representations in some radix system. Let us consider the definition of automatic sequences in [AS03] as a starting point.

Definition 4.1.1. Let \( N = (b, D, R) \) with \( D = \{0, 1, \ldots, b - 1\} \) and \( R = D^* \setminus \{0^+ D^*\} \) be a bijective canonical radix system for \( \mathbb{N} \). A sequence \((a_n)_{n \in \mathbb{N}}\) over an alphabet \( Y \) is \( b \)-automatic, if there exists a DFAO \( A \) with input alphabet \( D \) and output alphabet \( Y \) such that \( A(w) = a_n \) for all \( w \in R \), where \( n = \text{eval}_N(w) \).

It is apparent from this definition that the notion of automatic sequence holds with respect to some predetermined radix system in each given base. But which radix system should we choose when working with maps on, for example, the Gaussian integers instead of the natural numbers? We already know from theorem 3.3.1 that we cannot find a bijective canonical radix system for \( \mathbb{Z}[i] \) in every base.

Let us therefore state a general definition in terms of a radix system instead of a base, analogous to both definition 4.1.1 and the definition in [ACG+97].

Definition 4.1.2. Let \( N = (b, D, R) \) be a surjective radix system for a commutative ring \( S \), and \( Y \) an alphabet. A map \( f : S \to Y \) is called \( N \)-automatic, if there exists a DFAO \( A \) with input alphabet \( D \) and output alphabet \( Y \) such that \( A(w) = a_n \) for all \( w \in R \), where \( n = \text{eval}_N(w) \).

We may then also say that the DFAO \( A \) generates \( f \) under \( N \).

From the definition of automatic maps the following stability property is immediately clear.

Lemma 4.1.3. Let \( f : S \to Y \) be an \( N \)-automatic map, where \( N = (b, D, R) \). Then \( f \) is \( N' \)-automatic in every surjective \( N' = (b, D, R') \) with \( R' \subseteq R \).

In a similar vein we have that adding or removing unused digits maintains automaticity.

Lemma 4.1.4. Let \( f : S \to Y \) be an \( N \)-automatic map, where \( N = (b, D, R) \). Then \( f \) is \( N' \)-automatic in every \( N' = (b, D', R) \).

Proof. Let the DFAO \( A = (Q, q_0, D, \delta, Y, \pi) \) generate \( f \) under \( N \). Then the DFAO \( A' = (Q, q_0, D', \delta', Y, \pi) \) given by

\[
\delta'(q, d') = \begin{cases} 
\delta(q, d') & \text{if } d' \in D \\
q & \text{otherwise}
\end{cases}
\]

We have added 27
generates \( f \) under \( \mathcal{N}' \). Namely, since \( R \subseteq D^* \cap D'^* = (D \cap D')^* \), we have for any \( w \in R \) that
\[
A'(w) = \pi(\delta^*(q_0, w)) = \pi(\delta^*(q_0, w)) = A(w).
\]

\[\square\]

**Corollary 4.1.5.** Let \( R \subseteq D^* \). A map \( f: S \to Y \) is \((b, D, R)\)-automatic if and only if \( f \) is \((b, D \cup \{0\}, R)\)-automatic.

Other automaticity-preserving operations follow straight from automata theory.

**Lemma 4.1.6.** Let \( f: S \to Y \) be an \( \mathcal{N} \)-automatic map. Then

a) \( f \) is \( \mathcal{N} \)-automatic if and only if the maps
\[
f_y: S \to \{0, 1\},
\]
\[
f_y(s) = \begin{cases} 1 & \text{if } f(s) = y \\ 0 & \text{otherwise} \end{cases}
\]
are \( \mathcal{N} \)-automatic for all \( y \in Y \).

b) \( g \circ f \) is \( \mathcal{N} \)-automatic for any map \( g: Y \to Y' \);

c) \( f': S \to Y' \) is \( \mathcal{N} \)-automatic, where \( f' \) agrees with \( f \) except on a finite subset of \( S \);

**Proof.** Part [a] follows from corollary 2.2.8 and part [b] from 2.2.5. The last part follows from part [a], the fact that finite languages are regular, and lemma 2.3.3. \[\square\]

Now let us compare definition 4.1.2 with the definition of automatic maps in [ACG+97] \(^5\) which imposes the additional requirements on the radix system \( \mathcal{N} = (b, D, R) \) that

- \( 0 \in D \),
- \( R = D^* \setminus (0^+ D^*) \), and
- \( \mathcal{N} \) is bijective.

We are more lenient in these respects, and there is reason to. The following lemma together with lemma 3.3.8 shows that we cannot, for instance, create a \((1 + i)\)-automatic map for \( \mathbb{Z}[i] \) under their definition. In particular, automatic maps on the revolving system are excluded from consideration.

**Lemma 4.1.7.** Let \( \mathcal{N} = (b, D, R) \) be a bijective radix system for \( S \) with \( R = D^* \setminus (0^+ D^*) \). Then \( \mathcal{N} \) is a CRS.

**Proof.** Since every residue class of \( S/b \) is non-empty, we necessarily have that \( D \) includes a residue of each residue class in order for \( \mathcal{N} \) to be surjective. Suppose by way of contradiction that \( D \) contains two distinct residues \( d \neq 0 \) and \( d' \) of the same class. Then we can write \( d - d' = sb \) with \( 0 \neq s \in S \) uniquely, because \( \mathcal{N} \) is bijective. But then we get the contradiction that \( \mathcal{N} \) is not injective, since \( d \) is represented both by \( d \) and \( \text{repr}_{\mathcal{N}}(s)d' \). \[\square\]

\(^5\) The definition from section 2 that is used throughout the paper. In the last section, a more general definition of automatic maps is introduced briefly that is the same as we presented, except for the extra requirements that

- \( 0 \in D \), and
- \( R \) consists of \( \epsilon \) and exactly \( k \) representations for each non-zero \( s \in S \) for some positive integer \( k \).
On the other hand, we are being too lenient. Provided a surjective radix system exists for \( S \) at all, the following example shows that any conceivable map with finite range, no matter how unstructured, is automatic in some derived radix system with the same base. In particular, in case \( S = \mathbb{Z}[i] \) we know from section 3.3 that a surjective radix system exists for every base.

**Example 4.1.8.** Let \( f: S \rightarrow Y \) be any map into the alphabet \( Y = \{0, 1, \ldots, k\} \subseteq \mathbb{N} \), and \( N = (b, D, R) \) some surjective radix system for \( S \). We may assume that \( 0 \in D \), and that no representation has leading zeros. Then \( f \) is \((b, D, R')\)-automatic, where

\[
R' = \{o^{f(\text{eval}_N(w))}w \mid w \in R\}
\]

encodes the map in the leading zeros, because it is generated by for instance the DFAO from figure 10.

What we would like the notion of automaticity to convey, is some sense of structure in the map. Loosely speaking, it should not depend much on the specific radix system chosen. Perhaps the maps of our interest are precisely those whose automaticity can be characterized by a base alone.

**Definition 4.1.9.** A map \( f: S \rightarrow Y \) is \( b \)-automatic, if it is \( N \)-automatic in every surjective radix system \( N = (b, D, R) \) for \( S \).

Easy examples of \( b \)-automatic maps include the following type of periodic ones.

**Lemma 4.1.10.** Let \( N = (b, D, R) \) be a surjective radix system for \( S \), \( p \in S \) be such that \( S/p \) is finite, and \( g: S/p \rightarrow Y \) any map. Then the map

\[
f: S \rightarrow Y,
\]

\[
f(s) = g(s \mod p)
\]

is \( N \)-automatic.

**Proof.** We will construct a DFAO \( A = (Q, q_0, D, \delta, Y, g) \) that simply reduces the input modulo \( p \) as follows.

- \( Q = S/p \),
- \( q_0 = 0 \mod p \),
• \( \delta(q, d) = (qb + d) \mod p \).

It is straightforward to verify for any \( w \in D^* \) that \( \delta^*(q_0, w) = \text{eval}_\mathcal{N}(w) \mod p \), and therefore \( A(w) = f(\text{eval}_\mathcal{N}(w)) \).

However, unless your map is relatively simple, it is rather difficult to show that it is \( b \)-automatic. Now if we can show that for some radix system \( \mathcal{N} \), the class of \( \mathcal{N} \)-automatic maps coincides with the class of \( b \)-automatic maps, that would help us out greatly.

Let us therefore take a look at converting representations between radix systems in the same base, in a manner that can be implemented by an automaton. Since an automaton is able to simply guess a translation using non-determinism, the task reduces to deciding whether two words from different systems represent the same element. Furthermore, we will pad the representations conceptually with leading zeros in such a way that we may assume they have equal length.

So suppose \( \mathcal{N} = (b, D, R) \) and \( \mathcal{N}' = (b, D', R') \) are surjective radix systems for \( S \), and we are given \( w^R \in D^n \) and \( w'^R \in D'^n \) to check for some large \( n \) (reversed for convenience). In order for \( \text{eval}_{\mathcal{N}'}(w'^R) = \text{eval}_{\mathcal{N}}(w^R) \) to hold, we necessarily have for some \( c_1 \in S \) that

\[
w_0 = w'_0 + c_1 b.
\]

We will assume that \( b \) is no zero divisor, so \( c_1 \) is uniquely determined by \( w_0 \) and \( w'_0 \), and we learn that

\[
\text{eval}_{\mathcal{N}}(w_{n-1} \ldots w_1) + c_1 = \text{eval}_{\mathcal{N}'}(w'_{n-1} \ldots w'_1).
\]

Then analogously to the first step, we know there must exist a unique \( c_2 \in S \) such that

\[
w_1 + c_1 = w'_1 + c_2 b
\]

and so on. At the end of the inputs, we accept if and only if \( c_n = 0 \). This type of recurrence is perfectly suitable for implementation in a finite automaton, provided the carries \( c_0 = 0, c_1, \ldots, c_n \) are from a finite set for every possible pair of input words. That motivates the following technical condition.

We say two radix systems \( \mathcal{N} = (b, D, R) \) and \( \mathcal{N}' = (b, D', R') \) for a commutative ring \( S \) are \( C \)-convertible, if there exists a finite set \( C \subseteq S \) containing 0 such that for all \( d \in D, d' \in D' \) and \( c \in C \) the implication

\[
d + c \equiv d' \mod b \implies d + c = d' + \tilde{c} b \quad \text{for some } \tilde{c} \in C
\]

holds.

Let us briefly compare this definition with the technical notion in \[ACG+97\] that serves the same purpose. There, two bijective radix systems \( \mathcal{N} = (b, D, R) \) and \( \mathcal{N}' = (b, D', R') \) for a commutative ring \( S \) are \( C \)-linked, if there exists a finite set \( C \subseteq S \) containing 0 such that for all \( d \in D \) and \( c \in C \) there exist \( d' \in D' \) and \( \tilde{c} \in C \) such that

\[
d + c = d' + \tilde{c} b.
\]

We see that this notion is very similar to ours, but it appears to be less restrictive. However, recall from lemma \[4.1.7\] that the \( C \)-linked condition is only applied to CRSs, in which case both notions coincide.

Provided \( b \) is no zero divisor, a straightforward greedy procedure to find a smallest \( C \) for convertibility, if one exists at all, follows naturally from the definition. Starting with \( C_0 = \{0\} \), we simply follow the recurrence

\[
C_{n+1} = C_n \cup \{ \tilde{c} \in S \mid \tilde{c} b \in D - D' + C_n \}
\]

30
to get a monotone increasing sequence of finite sets \((C_n)_{n \in \mathbb{N}}\), that is bounded if and only if \(N\) and \(N'\) are convertible.

Fortunately, we do not have to worry about convertibility in case \(S = \mathbb{Z}\) or \(S = \mathbb{Z}[i]\) and we are working with a non-unary base. Here, the restriction to non-unary bases cannot be dropped, because we will see in a moment that not every two unary radix systems are convertible.

**Lemma 4.1.11.** Let \(S\) be either the ring \(\mathbb{Z}\) or \(\mathbb{Z}[i]\), and \(b \in S\) a base with \(|b| > 1\). Then any two radix systems \(N = (b, D, R)\) and \(N' = (b, D', R')\) are \(C\)-convertible, where we can take \(C\) arbitrarily large.

**Proof.** Let \(\Delta = \max_{d \in D} |\text{eval}_N(d)|\) and \(\Delta'\) similarly for \(D'\), and pick \(C = \{s \in S \mid |s| \leq \Gamma\}\) for any real \(\Gamma \geq \frac{\Delta + \Delta'}{|b| - 1}\). Then for any \(d \in D, d' \in D\) and \(c \in C\) with \(d + c \equiv d' \mod b\) we have that

\[
|d - d' + c| \leq |\text{eval}_N(d)| + |\text{eval}_{N'}(d')| + |c| \leq \Delta + \Delta' + \Gamma \leq |b| \Gamma,
\]

showing that \(\frac{d-d'+c}{b} \in C\). \(\Box\)

Now let us work towards showing that, under certain conditions, an \(N\)-automatic map is \(N'\)-automatic as well, which is what we were aiming for. We will need two lemmas, the first of which implements the subtask we discussed a moment ago, of checking whether two input words evaluate to the same element.

**Lemma 4.1.12.** Let \(N = (b, D, R)\) and \(N' = (b, D', R')\) be \(C\)-convertible surjective radix systems for the ring \(S\), where \(b\) is no zero divisor. Then the map

\[
f: (D \times D')^* \to \{0, 1\},
\]

\[
f((d_0, d_0') \ldots (d_{n-1}, d_{n-1}')) = \begin{cases} 1 & \text{if } \text{eval}_N(d_0 \ldots d_{n-1}) = \text{eval}_{N'}(d_0' \ldots d_{n-1}') \\ 0 & \text{otherwise} \end{cases}
\]

is \((b, D \times D', (D \times D')^*)\)-automatic.

**Proof.** We will construct an NFA \(A = (Q, I, D \times D', \delta, F)\) that generates \(f\) as follows, where we may assume that \(A\) reads its input starting with the least significant digit.

- \(Q = C\),
- \(I = \{0\}\),
- \(\delta(c, (d, d')) = \{\tilde{c} \in C \mid d + c = d' + \tilde{c}b\}\),
- \(F = \{0\}\).

Let us verify that \(A\) works as intended on every input \(w \in (D \times D')^*\). We will show by induction on the length of the input \(w = (d_0, d_0') \ldots (d_{n-1}, d_{n-1}')\) that

\[
0 \in \delta^*(c, w^R) \iff \text{eval}_N(d_0 \ldots d_{n-1}) + c = \text{eval}_{N'}(d_0' \ldots d_{n-1}'),
\]

from which the conclusion follows. It is definitely true if \(|w| = 0\), so let us assume it holds for \(w\) and prove it for \(w(d_n, d_n') \in D \times D'\) arbitrary. We have that \(0 \in \delta^*(c, (w(d_n, d_n'))^R)\) if and only if, by definition of \(\delta^*\), for some \(\tilde{c} \in C\)

\[
\tilde{c} \in \delta(c, (d_n, d_n')),
0 \in \delta^*(\tilde{c}, w^R),
\]

31
if and only if, by definition of $\delta$ and induction, for some $\tilde{c} \in C$

$$d_n + c = d'_n + \tilde{c}b,$$

$${\text{eval}}_N(d_0 \ldots d_{n-1}) + \tilde{c} = {\text{eval}}_{N'}(d'_0 \ldots d'_{n-1}),$$

if, by convertibility and cancellation of the factor $b$, and only if

$${\text{eval}}_N(d_0 \ldots d_{n-1}d_n) + c = {\text{eval}}_{N'}(d'_0 \ldots d'_{n-1}d'_n).$$

The other lemma we need makes sure that when we pad a representation with leading zeros, it remains a representation.

**Lemma 4.1.13.** Let $f : S \to Y$ be $N$-automatic, where $N = (b, D, R)$ is a regular radix system for $S$. Then $f$ is $N'$-automatic as well, where $N' = (b, D \cup \{0\}, 0^*R)$.

**Proof.** By corollary 4.1.5 we may assume that $0 \in D$, and by lemma 4.1.6a) we have that each $f_y$ is $N$-automatic, so let the DFA $A_y$ generate $f_y$ under $N$. Put otherwise, the language

$$\{w \in R \mid f_y({\text{eval}}_N(w)) = 1\} = R \cap L(A_y)$$

is regular. But then so is the language

$$\{w' \in 0^*R \mid f_y({\text{eval}}_{N'}(w')) = 1\} = 0^*\{w \in R \mid f_y({\text{eval}}_N(w)) = 1\}$$

$$= 0^*(R \cap L(A_y)),$$

showing that each $f_y$, and therefore $f$, is $N'$-automatic.

It is now relatively straightforward to tie everything together. As announced earlier, we build an automaton that simply guesses a translation of the input with the same length using artificial leading zeros, verifies whether the translation is correct, and if so, feeds the translation to a given target automaton and returns its response.

**Theorem 4.1.14.** Let $N = (b, D, R)$ and $N' = (b, D', R')$ be C-convertible surjective radix systems for the ring $S$, where $b$ is no zero divisor and $N$ is regular. Then every $N$-automatic map is $N'$-automatic.

**Proof.** By lemma 4.1.13 we may replace $N$ by $(b, D, 0^*R)$ with $0 \in D$, and by corollary 4.1.5 we may also assume that $0 \in D'$.

Let $f : S \to Y$ be an $N$-automatic map. According to lemma 4.1.6a) we have that each $f_y$ is $N$-automatic, so let the DFA $A_y = (Q, q_0, D, \delta, F)$ recognize the regular language $\{w \in 0^*R \mid f_y({\text{eval}}_N(w)) = 1\}$. Furthermore, let the DFA $(Q'', q_0', D \times D', \delta'', F'')$ generate the map from lemma 4.1.12. We will construct an NFA $A_y' = (Q', I', D', \delta', F')$ to generate $f$ under $N'$ as follows, where we assume that all automata read their input starting with the least significant digit.

- $Q' = Q \times Q''$,
- $I' = \{(q_0, q_0')\}$,
- $\delta'((q, q''), d') = \left\{ \delta(q, d), \delta''(q'', (d, d')) \right\} \mid d \in D$. 

32
• \( F' = \left\{ q' \in Q' \mid \delta^*_x(q', 0^n) \cap (F \times F'') \neq \emptyset \text{ for some } n \in \mathbb{N} \right\} \).

Let us verify that \( A'_y \) works as intended on every input \( w' \in R' \). By construction, we have that
\[
\delta^*_x((q, q''), w') \cap (F \times F'') \neq \emptyset
\]
if and only if for some \( w \in D^* \) with \(|w| = |w'|\)
\[
\delta^*(q, w) \in F, \quad \text{eval}_F(w) = \text{eval}_{F'}(w').
\]
We will use this remark to show that \( A'_y \) accepts \( w' \) if and only if \( f_y(\text{eval}_{F'}(w')) = 1 \). We have that \( \delta^*(q_0, w') \cap F' \neq \emptyset \) if and only if, by definition of \( F' \), for some \( q \in F \) and \( n \in \mathbb{N} \)
\[
(q, 0) \in \delta^*(q_0, (0^n w')^R),
\]
if and only if, by what we just remarked, for some \( n \in \mathbb{N} \) and \( w \in D^* \) with \(|w| = |0^n w'|\)
\[
\delta^*(q_0, w^R) \in F, \quad \text{eval}_F(w) = \text{eval}_{F'}(0^n w'),
\]
if and only if for some \( w \in D^* \) with \(|w| \geq |w'|\)
\[
\delta^*(q_0, w^R) \in F, \quad \text{eval}_F(w) = \text{eval}_{F'}(w'),
\]
if and only if for some \( w \in 0^* R \) with \(|w| \geq |w'|\)
\[
f_y(\text{eval}_{F'}(w)) = 1, \quad \text{eval}_F(w) = \text{eval}_{F'}(w'),
\]
if and only if for some \( w \in R \)
\[
f_y(\text{eval}_{F'}(w)) = 1, \quad \text{eval}_F(w) = \text{eval}_{F'}(w'),
\]
if and only if, by surjectivity of \( \mathcal{N} \),
\[
f_y(\text{eval}_{F'}(w')) = 1.
\]
This shows that each \( f_y \), and therefore \( f \), is \( \mathcal{N}' \)-automatic. \( \square \)

**Corollary 4.1.15.** Let \( S \) be either the ring \( \mathbb{Z} \) or \( \mathbb{Z}[i] \), and \( \mathcal{N} = (b, D, R) \) a regular surjective non-unary radix system. Then a map is \( \mathcal{N} \)-automatic if and only it is \( b \)-automatic.

As promised earlier, the following example illustrates that not every two regular unary radix systems in these rings are convertible. Therefore, we will leave these bases out of consideration when discussing \( b \)-automatic maps for \( \mathbb{Z}[i] \) in the remainder.

**Example 4.1.16.** Consider the Heaviside map
\[
f : \mathbb{Z} \to \{0, 1\},
\]
\[
f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}
\]
This map is easily seen to be \( \mathcal{N} \)-automatic, where \( \mathcal{N} = (b, D, R) \) is bijective for \( \mathbb{Z} \) with \( b = 1 \), \( D = \{-1, 1\} \) and \( R = (-1)^* 1^* \). See for instance figure [11]

However, \( f \) is not \( \mathcal{N}' \)-automatic, where \( \mathcal{N}' = (b, D, R') \) is surjective for \( \mathbb{Z} \) with \( R' = (-1)^* 1^* \).
For suppose that \( f \) is generated by the DFAO \( A = (Q, q_0, D, \delta, \{0, 1\}, \pi) \), and consider the input
Let us conclude this section with an example of a \((1+i)\)-automatic map for \(\mathbb{Z}[i]\). By corollary 4.1.15 we may just as well take the revolving system as our radix system underlying automaticity. As noted earlier, automatic maps on the revolving system do not fit in the standard framework of [ACG+97], so they especially accommodate for this system using the fact that every non-zero Gaussian integer has exactly four revolving representations. In contrast, we use that the revolving system is surjective and regular to support it.

Now we could for instance show that the indefinite extension of figure 7 induces a map

\[ f: \mathbb{Z}[i] \rightarrow \{0, \overline{1}, \overline{-1}, \overline{Q}, \overline{1}, \overline{Q}, \overline{0}, \overline{0}, \overline{0}, \overline{0}, \overline{0}, \overline{0}, \overline{0} \} \]

that is \((1+i)\)-automatic. Indeed, the part on revolving systems in section 3.3 provides all the necessary ingredients to do so. Given a revolving representation of a non-zero Gaussian integer, we can

- perform a Move step to deduce which line segment the representation corresponds to – this only requires the least significant non-zero digit and the least significant digit;
- perform a Turn step to deduce the direction of the bend following the line segment – this only requires the least significant non-zero digit and the immediately preceding digit (if any);
- determine the angle of rotation of the curve the line segment belongs to (i.e., the color) – this only requires the most significant non-zero digit.

All that we need to do to show that \(f\) is \((1+i)\)-automatic is to build an automaton that converts input to each of its four revolving representations using corollary 4.1.15 and extracts and combines the bits of information mentioned above. However, it is somewhat tedious to spell out everything in detail, so instead, we will take a prototypical example of an automatic map.
Theorem 4.2.1. A map automaton can be sped up or slowed down to read input in chunks of any constant size. A the DFAO positive integer $n$ automaticity to $b$ automaticity is invariant under such changes. Here we will investigate how we can relate Step for the appropriate copy of the dragon curve went from $z$ corresponding line segment is $z$ belongs to. One of the coordinates of a line segment in which direction it is traversed on the dragon curve it

Proof. $\Rightarrow$: We will show that $f$ is $\mathcal{N}'$-automatic for any surjective radix system $\mathcal{N}' = (b^n, D, R)$ for $S$. By definition, we may assume that $f$ is generated under $\mathcal{N} = (b, D \cup \{0\}, (0^{n-1}D)^*)$ by the DFAO $A = (Q, q_0, D \cup \{0\}, \delta, Y, \pi)$. Of course, $f$ is then also generated under $\mathcal{N}'$ by the DFAO $A' = (Q, q'_0, D, \delta', Y, \pi)$ given by

- $\delta'(q, d) = \delta^s(q, 0^{n-1}d)$.

$\Leftarrow$: We will show that $f$ is $\mathcal{N}'$-automatic for any surjective radix system $\mathcal{N}' = (b, D', R')$ for $S$. Let $L' = \bigcup_{k=0}^{n-1} D'^k$. By definition, we may assume that $f$ is generated under $\mathcal{N} = (b^n, D, D^*)$ with $D = \text{eval}_{\mathcal{N}'}(L')$ by the DFAO $A = (Q, q_0, D, \delta, Y, \pi)$, where we may assume that $A$ reads its input starting with the least significant digit. Then $f$ is also generated under $\mathcal{N}'$ by the DFAO $A' = (Q', q'_0, D', \delta', Y, \pi')$ given by

- $Q' = Q \times L'$.

Example 4.1.17. Let $\mathcal{N} = (b, D, R)$ be a bijective radix system for $S$ and consider the Thue-Morse map $f: S \to S/b$, $f(s) = \sum_{d \in \text{repr}_{\mathcal{N}}(s)} d \mod b$. Provided $S/b$ is finite, it is easy to show that $f$ is $\mathcal{N}'$-automatic in a way analogous to lemma 4.1.10.

Now let us take $S = \mathbb{Z}[i]$ and $\mathcal{N}$ the revolving system for $\mathbb{Z}[i]$ under the extra restriction that the representations of non-zero Gaussian integers have the unit $u$ as the least significant non-zero digit. In figure 12 we have visualized part of this map for $u = 1$. If you squint with your eyes a bit and/or look from a distance, it appears to exhibit some kind of distorted self-similarity that reminds of the dragon curve. What are we actually looking at here?

Recall from section 3.3 that we can generate a rotated copy of the dragon curve by alternating Turn and Move steps on the revolving representations, starting with a unit. The Turn step flips the parity of the number of non-zero digits, and the Move step adds a unit, so both change the residue class of the digit sum of the intermediate revolving representations modulo $1 + i$. Since we start with a unit and then Turn, we see that every Move step is performed on a revolving representation $w$ with $\sum_{d \in w} d \equiv 0 \mod b$. Put otherwise, the Thue-Morse map can tell from one of the coordinates of a line segment in which direction it is traversed on the dragon curve it belongs to.

Figure 12 can thus be explained as follows. If $N(z)$ is even, the other endpoint of the corresponding line segment is $z - i$, so the square centered on $z$ is gray if and only if the Move step for the appropriate copy of the dragon curve went from $z - i$ to $z$. Similarly, if $N(z)$ is odd, the same holds with $z - i$ replaced by $z - 1$.

4.2 Multiplicative independence of bases

In the previous section we discussed when automaticity of a map is preserved under a change of radix system in the same base, and we defined the $b$-automatic maps as those maps whose automaticity is invariant under such changes. Here we will investigate how we can relate $b$-automaticity to $b'$-automaticity.

The first result is standard and can also be found in [ACG+97]. In a way, it says that an automaton can be sped up or slowed down to read input in chunks of any constant size.

Theorem 4.2.1. A map $f: S \to Y$ is $b$-automatic if and only if it is $b^n$-automatic, for any positive integer $n$.

Proof. $\Rightarrow$: We will show that $f$ is $\mathcal{N}'$-automatic for any surjective radix system $\mathcal{N}' = (b^n, D, R)$ for $S$. By definition, we may assume that $f$ is generated under $\mathcal{N} = (b, D \cup \{0\}, (0^{n-1}D)^*)$ by the DFAO $A = (Q, q_0, D \cup \{0\}, \delta, Y, \pi)$. Of course, $f$ is then also generated under $\mathcal{N}'$ by the DFAO $A' = (Q, q'_0, D, \delta', Y, \pi)$ given by

- $\delta'(q, d) = \delta^s(q, 0^{n-1}d)$.

$\Leftarrow$: We will show that $f$ is $\mathcal{N}'$-automatic for any surjective radix system $\mathcal{N}' = (b, D', R')$ for $S$. Let $L' = \bigcup_{k=0}^{n-1} D'^k$. By definition, we may assume that $f$ is generated under $\mathcal{N} = (b^n, D, D^*)$ with $D = \text{eval}_{\mathcal{N}'}(L')$ by the DFAO $A = (Q, q_0, D, \delta, Y, \pi)$, where we may assume that $A$ reads its input starting with the least significant digit. Then $f$ is also generated under $\mathcal{N}'$ by the DFAO $A' = (Q', q'_0, D', \delta', Y, \pi')$ given by

- $Q' = Q \times L'$.
Figure 12: Part of the Thue-Morse map $f$ from example 4.1.17 on the restricted revolving system whose representations of non-zero Gaussian integers have 1 as the least significant non-zero digit. The centered square $z + [-\frac{1}{2}, \frac{1}{2}) + [-\frac{1}{2}, \frac{1}{2})i$ is colored gray if and only if $f(z) \equiv 1 \pmod{(1 + i)}$. 

36
\begin{itemize}
\item $q_0 = (q_0, \epsilon)$,
\item $\delta'((q, w'), d') = \begin{cases} 
(q, w'd') & \text{if } |w'd'| < n \\
\delta'(q, \text{eval}_{N'}(w'd')), \epsilon & \text{otherwise,}
\end{cases} \]
\item $\pi'((q, w')) = \pi(\delta'(q, \text{eval}_{N'}(w')))$,
\end{itemize}

that reads its input in reverse direction as well. Namely, $A'$ accepts $w' \in R'$ if and only if $A$ accepts a particular $w \in D'$ with $\text{eval}_{N'}(w') = \text{eval}_{N}(w)$.

That makes you wonder for what other bases $b'$ does $b$-automaticity coincide with $b'$-automaticity. In case $S = \mathbb{Z}$, Cobham’s deep theorem tells us there are no other $b'$ than we identified, because the characteristic map of $\{b^n \mid n \in \mathbb{N}\}$ is an example of a $b$-automatic map that never becomes periodic on the natural numbers.

**Theorem 4.2.2 (Cobham).** Let $a, b \in \mathbb{Z}$ be multiplicatively independent bases with $|a|, |b| \geq 2$. A map $f : Z \to Y$ is both $a$- and $b$-automatic if and only if both $f\mid_\mathbb{N}$ and $f\mid_{-\mathbb{N}}$ agree with periodic maps except on a finite subset.

But for $S = \mathbb{Z}[i]$, let alone for general rings, no one has shown an analogue of Cobham’s theorem yet, so we are at a loss. One might suspect for example that a map on the Gaussian integers is $\beta$-automatic if and only if it is $\beta'$-automatic. It turns out, however, that this only holds if we already have that $\beta^n = \beta^m$ for some positive integer $n$.

More precisely, we found that a map on the Gaussian integers is both $\alpha$- and $\beta$-automatic if and only if $\alpha^n = \beta^n$ for some positive integers $n$ and $m$. Hence, we can define the multiplicative dependence of two Gaussian bases in much the same way as for integer bases. We say two bases $\alpha, \beta \in \mathbb{Z}[i]$ are multiplicatively dependent if there exist positive integers $n$ and $m$ such that $\alpha^n = \beta^m$, and multiplicatively independent otherwise.

The remainder of this section is devoted to proving the aforementioned result. Before we are able to present the proof, we need to gather a few lemmas that are mostly related to the field of Diophantine approximation. In particular, we are interested in figuring out which complex numbers can be approximated arbitrarily well by fractions of the form $\frac{a}{b^n}$ with $n, m > 0$ and $\alpha$ and $\beta$ multiplicatively independent. For example, in case $\alpha$ and $\beta$ are natural numbers, it is relatively easy to prove that these fractions lie dense in the positive reals. But when $\alpha$ and $\beta$ are arbitrary Gaussian integers, the problem becomes much more difficult. In [HS03], a famous conjecture from number theory known as the Four Exponentials Conjecture\footnote{The Four Exponentials Conjecture states that for any two pairs $x_1, x_2 \in \mathbb{C}$ and $y_1, y_2 \in \mathbb{C}$ that are both linearly independent over the rational numbers, at least one of the four exponentials $e^{x_1 y_1}, e^{x_1 y_2}, e^{x_2 y_1}, e^{x_2 y_2}$ is transcendental.} had to be used as a resort to show that the given fractions with $\alpha = -a + i$ and $\beta = -b + i$ for some $a, b > 0$ lie dense in $\mathbb{C}$. Luckily, we can do without solving the entire problem of approximation, since we only need to be able to get close to some simple subset of the complex numbers.

First, we need a classical lemma that shows that the multiples of any real number must get arbitrarily close to an integer at some point, and we can even give a uniform upperbound on when that happens.

**Lemma 4.2.3 (Dirichlet).** Let $r$ and $\epsilon > 0$ be real numbers. Then there exist integers $n > 0$ and $m$ such that $|nr m| < \epsilon$. Moreover, we can take $n \leq N$, where $N = \lfloor \frac{1}{\epsilon} \rfloor$.

**Proof.** Let $N$ be as stated and consider the $N + 1$ fractional parts $\{0\}, \{r\}, \ldots, \{Nr\}$, where we use $\{x\}$ to denote $x - \lfloor x \rfloor$ as usual. Each remainder is contained in one of the $N$ intervals $[0, \frac{1}{N}), [\frac{1}{N}, \frac{2}{N}), \ldots, (\frac{N-1}{N}, 1)$, so by the pigeonhole principle there exist indices $0 \leq i < j \leq N$ such that $|ir j - \{ir\}| < \frac{1}{N} \leq \epsilon$. Hence, we can take $n = j - i$ and $m = nr - \{jr\} - \{ir\}$.

\[\Box\]
We can use the previous lemma to show how to approximate certain complex numbers by fractions $\frac{a^n}{\beta^m}$ with $n, m > 0$. It is even possible to upperbound both $n$ and $m$ uniformly as before when approximating 1, but since we have no use for it, we will not bother making this explicit.

**Lemma 4.2.4.** Let $\alpha, \beta \in \mathbb{Z}[i]$ with $|\alpha|, |\beta| > 1$, and $\epsilon > 0$ real. Then there exist integers $n, m > 0$ such that $\left|\frac{a^n}{\beta^m} - 1\right| < \epsilon$.

**Proof.** It suffices to show that we can find $z = \frac{a^n}{\beta^m}$ with both $e^{-\epsilon'} < |z| < e^{\epsilon'}$ and $|\text{Arg } z| < \epsilon'$ for some sufficiently small $0 < \epsilon' < \epsilon$, where $\text{Arg} : \mathbb{C} \setminus \{0\} \rightarrow (-\pi, \pi]$ denotes the usual principal value of arg.

Let $N = \lceil \frac{2\pi}{\epsilon'} \rceil$. By lemma 4.2.3 and $\log |\beta| \neq 0$ we can find integers $n' > 0$ and $m'$ such that

$$n' \log |\alpha| - m' \log |\beta| < \frac{\epsilon'}{N},$$

or, equivalently, find $z' = \frac{a^{n'}}{\beta^{m'}}$ with $n' > 0$ such that $e^{-\epsilon'/N} < |z'| < e^{\epsilon'/N}$.

By the same lemma, we can then also find integers $0 < k \leq N$ and $l$ such that

$$|k \text{ Arg } z' - l \cdot 2\pi| < \epsilon',$$

so we can take $z = z'^k$.

Finally, since we can take $n'$ arbitrarily large, and both $\log |\alpha|$ and $\log |\beta|$ are positive, we can ensure that $m' > 0$.

Of course, we can take $n$ and $m$ in the previous lemma arbitrarily large, so we can immediately derive the following consequence.

**Corollary 4.2.5.** There exist integers $n, m > 0$ such that $\left|\frac{a^n}{\beta^m} - z\right| < \epsilon$ for all complex $z$ of the form $\alpha^k \beta^l$ with $k, l \in \mathbb{Z}$.

The reason we were interested in approximation with fractions $\frac{a^n}{\beta^m}$ in the first place, and the reason we introduced compact numeration systems, is the following lemma. It basically shows how to extend certain representations in base $\beta$ to get a representation of a power of $\alpha$.

**Lemma 4.2.6.** Let $\mathcal{N} = (\beta, D, R)$ be a compact basic system for $\mathbb{Z}[i]$, $\alpha \in \mathbb{Z}[i]$, and $|\alpha|, |\beta| > 1$. Then for every $z \in \mathbb{Z}[i]$ of the form $\alpha^k \beta^l$ with $k, l \in \mathbb{N}$, there exists $z' \in \mathbb{Z}[i]$ and $m \in \mathbb{N}$ such that $|\text{repr}_\mathcal{N}(z')| \leq m$ and $z \beta^m + z' = \alpha^n$ for some $n \in \mathbb{N}$.

**Proof.** Let $\epsilon > 0$ be as in lemma 3.3.11. By the previous corollary, we can find integers $n, m > 0$ such that $\left|\frac{a^n}{\beta^m} - z\right| < \epsilon$, that is, $|\alpha^n - z \beta^m| < \epsilon |\beta|^m$. Therefore, by lemma 3.3.11, we can take $z' = \alpha^n - z \beta^m$, and the conclusion follows.

Finally, we are ready to show the result we announced. The idea behind the next theorem is to pick a ‘sparse’ map that is contained in one class of automatic maps, and reason by contradiction to show that it cannot be part of another. In order to pull this off, we will construct two distinct but ‘close’ input words that fools a supposed automaton into accepting them both, contradicting the map’s sparsity.
Theorem 4.2.7. Let $\alpha, \beta \in \mathbb{Z}[i]$ be multiplicatively independent with $|\alpha|, |\beta| > 1$. Then the class of $\alpha$-automatic Gaussian maps is different from the class of $\beta$-automatic Gaussian maps.

Proof. Without loss of generality we may assume that $N(\alpha), N(\beta) \geq 12$ by theorem 4.2.1. Furthermore, we may assume that $\mathcal{N}' = (\alpha, D', R')$ and $\mathcal{N} = (\beta, D, R)$ are compact basic numeration systems for $\mathbb{Z}[i]$ containing the digits 0 and 1 by theorem 3.3.7.

We will show that the map $f : \mathbb{Z}[i] \rightarrow \{0, 1\}$ given by

$$f(z) = \begin{cases} 1 & \text{if } z = \alpha^n \text{ for some } n \in \mathbb{N} \\ 0 & \text{otherwise} \end{cases}$$

is $\alpha$-automatic but not $\beta$-automatic.

The first part is clear. By corollary 4.1.15 we may choose to work in $\mathcal{N}'$, we have that $\text{eval}_{\mathcal{N}'}(w) = \alpha^n$ for $w \in R'$ if and only if $w = 1 \odot 0^n$, and the representations $1 \odot 0^*$ form a regular language.

Let us therefore focus on the second part, and suppose on the contrary that $f$ is $\beta$-automatic. By definition we may choose to work in $\mathcal{N}$, so let $A = (Q, q_0, D, \delta, F)$ be a DFA generating $f$. Consider the input word $v = 1 \odot 0^{|Q|} \in R$. By the pigeonhole principle, there exist indices $0 \leq s < s + l \leq |Q|$ such that $\delta^*(q_0, 1 \odot 0^s) = \delta^*(q_0, 1 \odot 0^{s+l})$. Hence, for any $w \in D^*$ we have that

$$f(\text{eval}_{\mathcal{N}'}(vw)) = f(\text{eval}_{\mathcal{N}'}(v0^kw)) \quad (2)$$

for every $k \in \mathbb{N}$.

Now let us successively identify representations $vw_{(0)}, vw_{(1)}, \ldots, vw_{(l)} \in R$ of powers of $\alpha$ by repeated extension. First, we find $w_{(0)} \in D^*$ such that $f(\text{eval}_{\mathcal{N}'}(vw_{(0)})) = 1$ by applying lemma 4.2.6 for $z = \text{eval}_{\mathcal{N}'}(v)$. Note for later that $w_{(0)}$ must contain a non-zero digit because $\alpha$ and $\beta$ are multiplicatively independent. Subsequently, given $vw_{(j-1)} \in R$ we find $w_{(j)} \in D^*$ by applying lemma 4.2.6 for $z = \text{eval}_{\mathcal{N}'}(vw_{(j-1)}0)$. In this way we made $vw_{(j-1)}$ a proper prefix of $vw_{(j)}$, so in particular we have that $|vw_{(j-1)}| < |vw_{(j)}|$.

By the pigeonhole principle there exist $0 \leq j' < j \leq l$ such that $|vw_{(j)}| = kl + |vw_{(j')}|$ for some positive integer $k$. Let $vw$ and $vw'$ be short for the representations $vw_{(j)}$ and $v0^kw_{(j')}$ respectively. Since $w_{(0)}$ contains a non-zero digit and is a prefix of both $w_{(j)}$ and $w_{(j')}$, these representations are distinct, and have different evaluations as $\mathcal{N}$ is bijective. By construction and by (2) we have that

$$f(\text{eval}_{\mathcal{N}'}(vw)) = 1 = f(\text{eval}_{\mathcal{N}'}(vw')),$$

so there exist distinct $n, n' \in \mathbb{N}$ such that $\text{eval}_{\mathcal{N}'}(vw) = \alpha^n$ and $\text{eval}_{\mathcal{N}'}(vw') = \alpha^{n'}$. Let us assume
that $n < n'$, the other case is analogous. But then we get the contradiction that

\[
|\alpha^n| < |\alpha^{n'}| - |\alpha^n| \quad \text{(since } |\alpha| > 2 \text{ and } n' > n) \\
\leq |\alpha^{n'} - \alpha^n| \quad \text{(triangle inequality)} \\
= |\text{eval}_N(vw) - \text{eval}_N(vw')| \\
= |\text{eval}_N(w) - \text{eval}_N(w')| \\
\leq |\text{eval}_N(w)| + |\text{eval}_N(w')| \quad \text{(triangle inequality)} \\
< 2|\beta|^{\lceil w \rceil} \quad \text{(by lemma 3.3.12)} \\
< 3|\beta|^{\lceil w \rceil} - |\text{eval}_N(w)| \quad \text{(by lemma 3.3.12)} \\
< |\beta|^{\lceil w \rceil + 1} - |\text{eval}_N(w)| \quad \text{(since } |\beta| > 3) \\
\leq |\text{eval}_N(v0^{\lceil w \rceil})| - |\text{eval}_N(w)| \quad \text{(since } |Q| \geq 1) \\
\leq |\text{eval}_N(vw)| \quad \text{(triangle inequality)} \\
= |\alpha^n|.
\]

\[
\square
\]

The consequences of the previous theorem are twofold. First, it settles the following open question posed in [ACG+97] negatively.

**Question 4.2.8.** Let $a \geq 2$ be an integer. Is every $(-a + i)$-automatic map on $\mathbb{Z}^i$ also $b$-automatic for some integer base $b \geq 2$?

This question comes up naturally if you only consider Gaussian bases that allow bijective canonical radix systems, and observing that $(-1 + i)$-automatic is the same as 2-automatic by theorem 4.2.1. It is now easy for us to answer the question because the previous theorem tells us that classes of automatic maps only coincide for multiplicatively dependent bases, and lemma 3.1.1 gives us that a power of a Gaussian integer is a natural number if and only if it is associated to a power of $1 + i$ times an integer.

Second, it reveals a part of the formulation of an analogue of Cobham’s theorem for $\mathbb{Z}^i$. The other part is likely to depend on the approximation qualities of the fractions $\frac{a^n}{\beta^m}$ with $n, m > 0$, and requires more investigation.
References


Index

N, 13
ε, 6
Z[i], 13
accept, 8
alphabet, 6
associate, 13
automatic map, 27
base, 17
base sequence, 16
complete residue system, 17
concatenation, 6
convertible, 30
CRS, see complete residue system
deterministic finite automaton, 7
deterministic finite automaton with output, 9
DFA, see deterministic finite automaton
DFAO, see deterministic finite automaton with output
digit set, 16
evaluation, 16
expansion, 17
final state, 7
fundamental domain, 14
generate, 10
initial state, 7
input alphabet, 7
language, 6
letter, 6
multiplicatively dependent, 37
multiplicatively independent, 37
NFA, see non-deterministic finite automaton
non-deterministic finite automaton, 8
norm, 13
numeration system, 16
  balanced, 24
  basic, 17
  bijective, 17
  canonical, 19
output alphabet, 9
output function, 10
prefix, 6
radix, 17
radix system, 17
recognize, 8
regular language, 11
representation, 17
representation set, 16
reversal, 6
state, 7
subword, 6
suffix, 6
symbol, 6
transition function, 7
unit, 13
word, 6
zero lead, 17