Automata on Shift Spaces



A dissertation submitted towards the degree
Doctor of Natural Sciences
of the Faculty of Mathematics and Computer Science
of Saarland University

by

Mihir Vahanwala

Saarbrücken, 2026

Day of Colloquium	TBA 2027
Dean of the Faculty	Prof. Dr. TBC
Chair of the Committee	Prof. Dr. TBA
Reporters	Prof. Dr. Joël Ouaknine Prof. Dr. TBA Prof. Dr. TBA Prof. Dr. TBA
Academic Assistant	Dr. TBA

This page is intentionally left blank.

Abstract

A word is a sequence of symbols drawn from a finite set, and is often a device to represent the evolution of a (discretised) system. Words are central to both symbolic dynamics and verification. In the former discipline, words that represent the same system constitute a *shift space*, and are typically studied for their combinatorial structure. In verification, a quintessential objective is to certify that a system satisfies a logical specification by showing that its representative words are accepted by an automaton. In this thesis, we bridge the two perspectives through the following contributions.

1. We describe how the action of a deterministic finite transducer on an infinite word preserves the structure of the word. If the input word x has a recurrence property, then this property is preserved for the output word $\mathcal{A}(x)$; using the ergodic theory of shift spaces, we identify a large class of words x, including *Sturmian words*, such that $\mathcal{A}(x)$ is guaranteed to admit factor frequencies.

2. We show that we can determine whether an automaton accepts a word by reading its so-called *S-adic construction*. This allows us, e.g., to decide, given an automaton $\mathcal{A}$, whether $\mathcal{A}$ accepts all Sturmian words, or all d-letter *Arnoux-Rauzy words*.

3. As applications, we study *Monadic Second-Order (MSO)* logic, and show that the MSO theory of the natural numbers remains decidable when expanded with the following sets of predicates: (i) powers of 2 and 3; (ii) powers of $2, 3$, and 5, subject to Schanuel's conjecture; (iii) perfect squares and powers of 2, subject to the binary expansion of $\sqrt{2}$ being a *disjunctive word*.

Zusammenfassung

Ein Wort ist eine Folge von Symbolen aus einer endlichen Menge und dient häufig dazu, die Entwicklung eines diskretisierten Systems darzustellen. Wörter spielen sowohl in der symbolischen Dynamik als auch in der Verifikation eine zentrale Rolle. In der symbolischen Dynamik bilden Wörter, die dasselbe System repräsentieren, einen Schiftraum und werden typischerweise hinsichtlich ihrer kombinatorischen Struktur untersucht. In der Verifikation besteht ein grundlegendes Ziel darin, nachzuweisen, dass ein System eine logische Spezifikation erfüllt, indem gezeigt wird, dass die das System repräsentierenden Wörter von einem Automaten akzeptiert werden. In dieser Dissertation führen wir diese beiden Perspektiven anhand der folgenden Beiträge zusammen:

1. Wir beschreiben, inwiefern die Wirkung eines deterministischen endlichen Transduktors auf ein unendliches Wort dessen Struktur erhält. Besitzt das Eingabewort x eine Rekurrenzeigenschaft, so bleibt diese auch im Ausgabewort $\mathcal{A}(x)$ erhalten. Mithilfe der Ergodentheorie von Schifträumen identifizieren wir eine große Klasse von Wörtern x, darunter Sturmsche Wörter, für die garantiert ist, dass $\mathcal{A}(x)$ Faktorfrequenzen besitzt.

2. Wir zeigen, dass sich durch das Lesen der sogenannten S-adischen Konstruktion eines Wortes bestimmen lässt, ob dieses von einem Automaten akzeptiert wird. Dadurch können wir beispielsweise für einen gegebenen Automaten $\mathcal{A}$ entscheiden, ob $\mathcal{A}$ sämtliche Sturmsche Wörter oder sämtliche Arnoux-Rauzy-Wörter über einem Alphabet mit d Buchstaben akzeptiert.

3. Als Anwendungen untersuchen wir die monadische Logik zweiter Stufe (MSO-Logik) und zeigen, dass die MSO-Theorie der natürlichen Zahlen entscheidbar bleibt, wenn sie um die folgenden Mengen von Prädikaten erweitert wird: (i) Potenzen von 2 und 3; (ii) Potenzen von $2, 3$ und 5 unter der Voraussetzung der Schanuel-Vermutung; (iii) Quadratzahlen und Potenzen von 2 unter der Voraussetzung, dass die Binärentwicklung von $\sqrt{2}$ ein disjunktives Wort ist.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit *Automata on Shift Spaces* selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Declaration of original authorship

I hereby declare that this dissertation *Automata on Shift Spaces* is my own original work except where otherwise indicated. All data or concepts drawn directly or indirectly from other sources have been correctly acknowledged. This dissertation has not been submitted in its present or similar form to any other academic institution either in Germany or abroad for the award of any other degree.

Saarbrücken, November 2026

Mihir Vahanwala

Acknowledgements

I am grateful to my advisor, Joël Ouaknine, for the immense academic freedom I enjoyed while working in the Foundations of Algorithmic Verification group he leads. It is through this collaborative environment that I found mentors and role models in Valérie Berthé, whose direction has been profoundly instrumental to the science in this thesis, and [James] Ben Worrell. I owe the impetus that started my professional academic career to the belief Parosh Abdulla had in me when I was an undergraduate.

I would also like to pay a solemn tribute to the memory of Florian Luca, for I was fortunate to have witnessed his mathematical brilliance and high spirits that inspired a generation.

I am grateful to the following individuals who have contributed to my progress primarily through their kind service as examiners at various junctures during my PhD: Holger Hermanns (Qualifying Exam), Georg Zetzsche (Qualifying Exam, Thesis Proposal), (PhD Exam TBA), (PhD Exam TBA), and (PhD Exam TBA).

Much of my science has been produced working closely with Toghrul [Karimov] and Joris [Nieuwveld], and I am grateful for the collaboration and camaraderie. I have been privileged to exchange ideas with Laurent [Bartholdi], Christel [Baier], and Martina [Maggio], and to count Dominique [Perrin] and Herman [Goulet-Ouellet] among my co-authors, each of whom I thank ever so much. I would also like to acknowledge the graciousness of those who hosted my research visits: Mrinal [Kumar] at TIFR, Mumbai, [B.] Srivathsan at CMI, Chennai, and Bastián [Espinoza] and Manon [Stipulanti] in Liège.

While pursuing my doctoral degree, I was employed at the Max Planck Institute for Software Systems in Saarbrücken, Germany, as a *wissenschaftlicher Mitarbeiter* (scientific employee). I would like to thank everyone at MPI-SWS for creating a stimulating scientific environment and a wonderful workplace. I am especially grateful to Annika, who managed my onboarding, Lena and Taieb, who manage our research group's bureaucracy, Gretchen and Mary-Lou for their work in the Graduate Studies Office, Claudia [Hesse] for travel support, Claudia [Richter] and Ruth for the help they gave with immigration bureaucracy, and indeed the entire office team across both locations of MPI-SWS.

For some periods during my PhD, I worked part-time as a tutor for courses at Saarland University, and I thank Bernd Finkbeiner and Markus Bläser for immensely fulfilling opportunities.

I would like to mention MPI-SWS colleagues with whom I have shared the journey in one form or another — Piotr, and indeed all the members of our Foundations of Algorithmic Verification group, which grows in strength and ambition; Ashwani, Bala, Corto, Henry, Irmak, Sathiya, Satya, to name but few of those with shared research interests; Ayce, Laila, and Mahsa, with whom I shared my office space; Soumi and Abhisek, who were helpful and friendly neighbours.

It has been a privilege to travel to conferences and workshops and connect with peers — I have often found myself enriched by technical and social conversations with people like Clara, Rémy, and Ruiwen.

I conclude my vote of thanks with gratitude for the love that bridges a continent[1] to sustain me — that of my parents, Severina and Jayesh, and that of my dear Devina.

[1] It is Eurasia.

To my grandfather Joseph, whose mathematical dreams I am privileged to pursue

हा प्रबंध मी माझे आजोबा जोसेफ यांना समर्पित करू इच्छितो. त्यांच्या गणिताच्या स्वप्नांचा पाठपुरावा करणे माझे सौभाग्य आहे.

Contents

Introduction **1**

1 Preliminaries **7**
1.1 Words . . . 7
1.1.1 Morphic Words . . . 8
1.1.2 Sturmian Words . . . 9
1.1.3 S-adic Words . . . 12
1.1.4 Recurrent Words . . . 19
1.1.5 Effective Uniform Recurrence . . . 23
1.2 Automata . . . 27
1.2.1 Monadic Second-Order Logic . . . 28
1.2.2 Equivalence with Semigroups . . . 32
1.2.3 The Krohn–Rhodes Framework . . . 35
1.3 Shift Spaces . . . 37
1.3.1 Factor Languages . . . 38
1.3.2 Measures . . . 39

2 Automata on Recurrent Words **41**
2.1 Preservation of Recurrence . . . 41
2.2 Prefix-Independent Verification . . . 47
2.3 Effective Recurrence Revisited . . . 51

3 Preservation of Factor Frequencies **56**
3.1 Factor Frequencies and Genericity . . . 56
3.2 Unique Ergodicity is Perishable . . . 59
3.3 The Morphic Case . . . 63
3.4 Preservation of Boshernitzan's Condition . . . 66
3.4.1 The Case with Reset Letters . . . 67
3.4.2 The Case of Permutation Automata . . . 68

3.4.3 Further Technical Remarks 71

4 Automata on S-adic Words **74**
4.1 Structure Lemmas for Directive Sequences 74
4.2 Congeniality and Congruence 77
4.3 Partial Quotients 79
4.4 Periodic Generating Sequences 81
4.5 Perspective and Future Work 84

5 The Monadic Theories of Arithmetic Predicates **87**
5.1 Reduction to Order Words 88
5.2 Automata on Almost-Periodic Order Words 94
5.2.1 Uniform Recurrence of Cutting Sequences 95
5.2.2 Effective Uniform Recurrence of Cutting Sequences 98
5.3 Automata on High-Entropy Order Words 102
5.4 Discussion 104

Concluding Perspective **106**
Expressive Power and Preservation Theorems 106

Bibliography **109**

Index **115**

Introduction

Bee Gees, 1968

It's only words,
And words are all I have
To take your heart away.

We inhabit a universe that evolves in myriad ways, and it is our nature to seek order and predictability. Consider, for instance, the basic ancient endeavour of time-keeping. We had at our immediate disposal the cycles of day and night, and of the seasons; with a little more sophistication, we observed the phases of the Moon. We now know that these phenomena are driven by *continuous dynamical systems*: the rotation of the Earth about its axis, the revolution of the Earth about the Sun, the revolution of the Moon around the Earth. The path to this knowledge and understanding, however, was one forged by making *discrete* observations: sightings of the full moon are roughly 30 sunsets apart, the longest and warmest days occur roughly 12 lunar cycles apart, and so on. Since antiquity, we have had the mathematical instinct to represent a *trace* of observations of a system using a discrete sequence of symbols. Symbolic dynamics and verification are two modern disciplines that formalise this technique: in today's parlance, we call the (finite) set of symbols an *alphabet*, and the discrete sequences *words*.

When we discretise systems that consist of multiple cycles, one challenge is that these cycles do not necessarily synchronise, giving us aperiodic words. As an example, consider the relation between an average day and the average tropical year. One may write an infinite word over the alphabet $\{0, 1\}$ as follows (see Fig. 1). Distinguish a point on the Earth's orbit, e.g., the point that defines the northern spring equinox. We write a letter each midnight: we write 1 if the Earth is at the distinguished point or has crossed it during the previous day, and 0 otherwise. Since the average tropical year is around 365.2422 days long, there will be either 364 or 365 intervening 0-s between consecutive appearances of 1, depending on how late in the day the moment of equinox occurred.

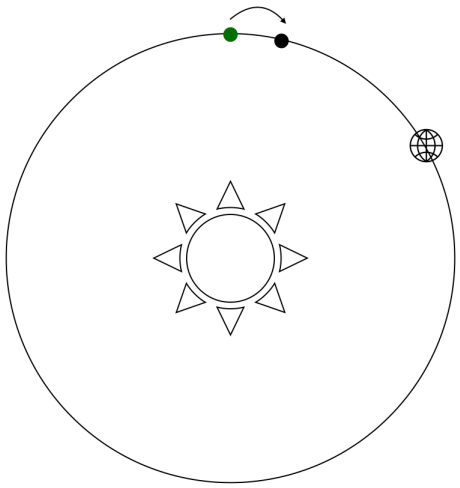

Figure 1: We assume the Earth revolves around the Sun in a perfectly circular, constant orbit. In one day, the Earth revolves the angle between the green and black points. The green point represents the point at which the northern spring equinox occurs. We write a letter every midnight: 0 if there was no equinox, and 1 if the moment of equinox is at present, or occurred after midnight during the previous day. The condition to write the letter 1 is equivalent to the Earth being in the arc between the green and black points, including the green point, at midnight.

Unless the number of days in an average tropical year is rational, we have constructed an aperiodic word; in fact, this would be a *Sturmian word*. The kind of process described above is indeed how lunar months are defined, e.g., declaring the sighting of a new crescent moon to mark the start of a new month. In practice, we do get aperiodic words, but not the mathematically predicted Sturmian words as celestial orbits are slightly elliptical, and human observation can be imprecise.

(The standard Gregorian calendar, meanwhile, was designed for simplicity and administrative convenience to be periodic with a 400-year cycle. It does not synchronise with the Earth's revolution: the average calendar year is exactly 365.2425 days long, a few seconds longer than the average tropical year. One might, after learning about Sturmian words in Sec. 1.1.2, be tempted to define a highly accurate calendar based on the *continued fraction expansion* of the length of the average tropical year. The astronomers of Pope Gregory XIII, doubtless mathematically inclined, resisted this temptation in favour of practicality.)

Words are also exemplified by decimal representations of numbers, e.g., the digits of the circle constant π define the word $314159265358979\cdots$ over the alphabet $\{0, 1, \ldots, 9\}$. The following discrete dynamical system generates the decimal representation of a real number $\zeta_0 \in [0, 1)$: partition the unit interval into the subintervals $[0, 0.1), [0.1, 0.2), \ldots, [0.9, 1)$, respectively named $0, 1, \ldots, 9$. At the n-th step, record the interval that ζ_n is in, and define ζ_{n+1} to be the fractional part of $10 \cdot \zeta_n$. Of course, we may consider other bases to represent numbers: e.g., the digits of $\sqrt{2}$ in binary define the word $10110101000\cdots$.

Symbolic dynamics is the study of words that form traces of dynamical systems such as the ones described above. Lind and Marcus [52, Preface] attribute the foundations to Hadamard [42], who, in 1898, introduced the quintessential elements of the discipline while studying more complicated systems called geodesic flows on surfaces of negative curvature.

A central object in symbolic dynamics is the *shift space*, which is the set of all infinite words that do not contain a given set of *forbidden factors*. (A forbidden factor is a finite sequence of letters that may not occur as a contiguous subword of the words under consideration.) In the ensuing decades, it was discovered by mathematicians such as Morse, Hedlund, and others, that in addition to the ones Hadamard studied, several other dynamical systems are also such that the respective sets of their symbolic traces constitute shift spaces. Today, shift spaces are considered interesting in their own right, and the study of their attendant word combinatorics motivates active research.

Words were found to play a central role in logic and verification by Büchi, whose seminal 1962 paper [18] showed that the *Monadic Second-Order (MSO)* theory of the structure $\langle \mathbb{N}; < \rangle$ is decidable. The influence of Büchi's work lies in the following verification paradigm: should a system designer wish to verify that a trace satisfies a specification which can be formulated in MSO logic, they construct an equivalent *automaton* and determine whether it *accepts* the trace.

In 1966, shortly after Büchi's work, Elgot and Rabin [35] used automata theory to show that the MSO theory of the expansion $\langle \mathbb{N}; <, P \rangle$ is decidable when P is a unary predicate with "arithmetic" origins, e.g., the set of squares, the set of powers of two, the set of factorial numbers, and so on. The key property of the sequences giving rise to these sets is that they are ultimately periodic modulo any integer divisor. The techniques and results were later generalised by Carton and Thomas [20] through the notion of *(effective) profinite ultimate periodicity*.

The natural question is whether expansions with multiple predicates have decidable MSO theories. Examples of such structures include:

1. $\langle \mathbb{N}; <, \{2^n \mid n \in \mathbb{N}\}, \{3^n \mid n \in \mathbb{N}\} \rangle$,
2. $\langle \mathbb{N}; <, \{2^n \mid n \in \mathbb{N}\}, \{3^n \mid n \in \mathbb{N}\}, \{5^n \mid n \in \mathbb{N}\} \rangle$,
3. $\langle \mathbb{N}; <, \{2^n \mid n \in \mathbb{N}\}, \{n^2 \mid n \in \mathbb{N}\} \rangle$,
4. $\langle \mathbb{N}; <, \{n! \mid n \in \mathbb{N}\}, \{n^2 \mid n \in \mathbb{N}\} \rangle$.

The developments in this thesis are applicable to the first three examples. Observe, meanwhile, that the fourth example subsumes the Brocard-Ramanujan problem of determining whether there are solutions other than $n \in \{4, 5, 7\}$ to the equation $n^2 = m! + 1$. The question above is thus open in general.

This is a challenging research question, not least because numerically sophisticated dynamics may underlie such expanded structures. It is only recently [12] that substantial progress has been reported: using automata theory and Baker's (Fields-medal-winning) theory of linear forms of logarithms, we reduce, e.g., the problems of deciding the first three theories respectively to the problems of deciding whether a given automaton accepts the following words, which have a distinctly symbolic-dynamical nature:

1. the Sturmian word $2322323223\cdots$, which records the order in which the powers of 2 and 3 appear,

2. the *cutting sequence* $2325232532\cdots$, which records the order in which the powers of $2, 3$, and 5 appear,

3. the word $10110101000\cdots$ formed by the digits of the binary expansion of $\sqrt{2}$.

We shall, of course, duly return to the solutions of these reduced problems. It is worth noting that, at this point, we may ask the broader question that forms the subject of this thesis:

How do automata run on shift spaces?

Semënov's study of *almost-periodic* words (see, e.g., [58]) is instrumental in answering this question. Chap. 2, based on [14], adds insights to, and extends such results. We show that recurrence properties of a word x also hold for the output $\mathcal{A}(x)$ when x is input to a deterministic finite transducer $\mathcal{A}$.

Chap. 3, also based on [14], is the measure-theoretic counterpart to the word-combinatorial chapter that precedes it. The motivating question is to understand the set of words x for which the transducer output $\mathcal{A}(x)$ is guaranteed to *admit factor frequencies*. This is a profound question, and one intrinsically connected to symbolic dynamics and ergodic theory: we show that a word admits factor frequencies if and only if it is *generic* for an *invariant measure* on the shift it generates. We focus on cases where we can prove that the shift generated by $\mathcal{A}(x)$ is *uniquely ergodic*, i.e., it has only one invariant measure for which *all* points are generic: *Boshernitzan's condition* is key for our techniques. As alluded to in the abstract, Sturmian words satisfy Boshernitzan's condition.

Chap. 4, which is based on [13], marks a return to the perspective of automata as recognisers of ω-regular languages. Its main results have the following form: given an ω-regular language L, we can compute an ω-regular language L' such that a word $x \in L$ if and only if its *S-adic directive sequence* $\boldsymbol{\sigma} \in L'$. The point is that directive sequences often elucidate structural information about a word, e.g., directive sequences of Sturmian words indicate their *slopes* and *intercepts*.

Chap. 5, the final technical chapter of the thesis, is based on [12], and applies the developments of the preceding chapters to give (sometimes conditional) decision procedures for the automaton acceptance problems of order words such as the ones mentioned above. We thus obtain (conditional) decidability results for the corresponding MSO theories.

The technical preliminaries recorded in Chap. 1 lay the foundations of the developments in this thesis.

Declaration. This thesis is based on original research that I have conducted in collaboration with the authors of [12, 13, 14]. I have played integral roles in the technical development of [12] with Toghrul Karimov and Joris Nieuwveld, and the technical development of [13] with Toghrul Karimov. I have led the technical development of [14].

The German version of the abstract was obtained by translating the original English version using a generative AI tool. I thank Wolfgang Steiner for reviewing the translation.

Chapter 1

Preliminaries

This thesis is based on the interplay between word combinatorics, ω-regular languages, and symbolic dynamical systems. We introduce the key concepts in this chapter, giving basic definitions and sharing proofs of some standard properties. As usual, $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ respectively denote the sets of natural numbers, integers, rational numbers, real numbers, and complex numbers. We follow the computer-scientific convention of 0 being a natural number.

1.1 Words

An *alphabet* Σ is a finite nonempty set of symbols known as *letters*. We shall typically assume $|\Sigma| > 1$ unless otherwise stated. Given an alphabet, we call a sequence of its letters a *word* over the alphabet. The *length* of a finite word u is defined in the natural way, and is denoted $|u|$; the length of infinite words is taken to be ∞. The notation Σ^ℓ denotes the set of all length-ℓ words over the alphabet Σ; we use ε to denote the empty word in Σ^0. We use Σ^*, Σ^+, and Σ^ω to respectively denote the sets of finite words, finite nonempty words, and infinite words (indexed by the ordinal ω) over the alphabet Σ. We define Σ^∞ as $\Sigma^+ \cup \Sigma^\omega$. In this thesis, letters will typically be referred to by a, b, c or $0, 1, 2, \ldots$, finite words by u, v, w, and arbitrary words by x, y, z.

For $i \in \mathbb{N}$, the notation $x(i)$ refers to the letter in the i-th position of the word x. For a length $\ell \in \mathbb{N}$, the notation $x(i, i+\ell)$ refers to the subword of x starting at index i and having length ℓ. For example, if $x = abaaba$, then $x(1,3) = ba$, and $x(1,1) = \varepsilon$. If x is an infinite word, then $x(i, \infty)$ denotes the suffix of x starting at index i. We say that a word x is *effective* if we can compute $x(i)$ given any index i.

We equip $\Sigma^* \cup \Sigma^\omega$ with the usual *product topology*, and define a metric as follows. Given distinct words x, y, let n be the index of the first position where x and y differ.

The distance between x and y is then 2^{-n}. If x is a finite word of length ℓ that is a prefix of y, then the distance between x and y is $2^{-\ell}$.

A finite word $u \in \Sigma^\ell$ is a *factor* of x if for some index i, we have $x(i, i+\ell) = u$. We call the index i an *occurrence* of u in x. The *frequency* of the factor u in x is defined as

$$f_x(u) = \frac{|\{i \mid x(i, i+\ell) = u\}|}{|x| - |u| + 1} \tag{1.1}$$

if x is finite. In our above example where $x = abaaba$, we have $f_x(ba) = 2/5$, because ba occurs in 2 of the 5 possible indices that a length-2 factor could start in. If x is infinite, then the frequency of u in x is defined as

$$f_x(u) = \lim_{N\to\infty} \frac{1}{N} |\{i < N \mid x(i, i+\ell) = u\}| \tag{1.2}$$

provided the limit exists. For example, if $x = (aab)^\omega$, then $f_x(a) = 2/3$. We say that x admits *computable factor frequencies* if moreover, given any finite word u and precision δ, we can compute a rational number ν such that $|f_x(u) - \nu| < \delta$.

For a word x, we define its *factor language* $\mathcal{L}(x) \subseteq \Sigma^*$ to be the set of its factors. We declare $\mathcal{L}_\ell(x)$ as shorthand for $\mathcal{L}(x) \cap \Sigma^\ell$, i.e., the set of length-ℓ factors of x. We define the *factor-complexity function* $p_x : \mathbb{N} \to \mathbb{N}$ as $p_x(\ell) = |\mathcal{L}_\ell(x)|$.

Lemma 1.1.1. *For an effective word x, having a recursive factor language $\mathcal{L}(x)$ is equivalent to having a computable factor-complexity function p_x.*

Proof. Indeed, with oracle access to $\mathcal{L}(x)$, one can enumerate all the elements of $\mathcal{L}_\ell(x)$ given any ℓ; on the other hand, if one is given $p_x(\ell)$, then one can compute a long enough prefix of x until the required number of length-ℓ factors have been enumerated. □

1.1.1 Morphic Words

Let Σ, Γ be alphabets. A *substitution* is a map $\Sigma \to \Gamma^*$, and is extended to Σ^* and Σ^ω in the obvious way, i.e., concatenation. A substitution $\sigma : \Sigma \to \Sigma^+$ is said to be *primitive* if there exists a positive integer m such that for every letter $a \in \Sigma$, all letters occur in $\sigma^m(a)$. A substitution is said to be *non-erasing* if there does not exist a letter a such that $\sigma(a) = \varepsilon$.

A word $x \in \Sigma^\omega$ (we assume without losing generality that x contains all the letters in Σ) is called *substitutive* if there exists a non-erasing substitution $\sigma : \Sigma \to \Sigma^+$, and a letter a such that $\sigma(a) \in a\Sigma^+$ and $x = \lim_{n\to\infty} \sigma^n(a)$. If σ is primitive, then x is said to be a *primitive substitutive word*.

As an example, the Fibonacci word $x_{\mathsf{Fib}} = 0100101001001\cdots$ is primitive substitutive, and is the fixed point of the substitution that maps 0 to 01 and 1 to 0. As another example, the Thue-Morse word $x_{\mathsf{TM}} = 01101001100101\cdots$ is also primitive substitutive as it is a fixed point of the substitution that maps 0 to 01, and 1 to 10.

Yet another example is the word x_{pow3}, which is the word indicating powers of 3: $x(0) = 2$, and for $i \geq 1$, we have $x(i) = 1$ if i is a power of 3, and 0 otherwise. This word is a fixed point of the substitution that maps 0 to 000, 1 to 100, and 2 to 210. This word is substitutive, but not primitive substitutive.

A word $y \in \Gamma^\omega$ is called *morphic* if there exists a substitutive word $x \in \Sigma^\omega$, and a substitution $\tau : \Sigma \to \Gamma^*$ such that $y = \tau(x)$. If x is primitive substitutive, then y is said to be *primitive morphic*.

Morphic words are represented by σ, τ, and a letter a such that $\sigma(a)$ prolongs a (i.e., begins with a and is more than one letter long). We can effectively convert any representation into one such that σ is non-erasing and τ is a coding, i.e., a map from Σ to Γ [3, Sec. 7.7]. This is a result originally due to Cobham; see also [31, Thm. 3.8] (Thm. 9 in the arXiv version), [46], [32, Prop. 17] for other technical expositions.

Finally, we record the following lemma, whose proof can be found in [3, Thm. 7.6.1].

Lemma 1.1.2. *Let x be a morphic word given by substitutions σ, τ. For any $n \geq 1$, the suffix $x(n, \infty)$ is also a morphic word, and we can compute the substitutions σ', τ' that describe it.*

1.1.2 Sturmian Words

Let $\eta \in (0, 1)$ be an irrational number. The *characteristic Sturmian word* x_η with *slope* η is an infinite word over the alphabet $\{0, 1\}$. The letter at index i is given by

$$x_\eta(i) = \lfloor (i+2)\eta \rfloor - \lfloor (i+1)\eta \rfloor, \tag{1.3}$$

where $\lfloor \zeta \rfloor$ denotes the largest integer that does not exceed ζ. (Dually, we define $\lceil \zeta \rceil$ to be the smallest integer that is at least ζ.)

For example, the Fibonacci word is the characteristic Sturmian word with slope $1/\phi^2$, where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio.

A characteristic Sturmian word can be viewed as the coding of a discrete rotation on the 1-torus. Indeed, we identify the 1-torus with the interval $[0, 1)$, partition it as $[0, 1-\eta)$ and $[1-\eta, 1)$, take $\zeta_0 = \eta$ as our initial point, define $\zeta_{i+1} = \zeta_i + \eta$, and define the i-th letter of the coding to be 0 in $\zeta_i \in [0, 1-\eta)$, and 1 otherwise.

More generally, a Sturmian word can have an *intercept* $\chi \in [-\eta, 1-\eta]$, and is given by one of the following:

$$x_{\eta,\chi}(i) = \lfloor (i+2)\eta + \chi \rfloor - \lfloor (i+i)\eta + \chi \rfloor \text{ for all } i, \tag{1.4}$$

$$x_{\eta,\chi}(i) = \lceil (i+2)\eta + \chi \rceil - \lceil (i+i)\eta + \chi \rceil \text{ for all } i. \tag{1.5}$$

The distinction between using floor (obtaining what is referred to as a *lower mechanical word*) and ceiling (referred to as an *upper mechanical word*) is irrelevant unless $\chi \in \mathbb{Z} + \mathbb{Z}_{\leq -1}\eta$, which is the only case where an integer will be input (at most twice) to a rounding function.

A Sturmian word can also be viewed as the coding of a discrete rotation similarly to the characteristic Sturmian word. In the general case, the starting point is arbitrary, and the partition can either be $\{[0, 1-\eta), [1-\eta, 1)\}$ or $\{(0, 1-\eta], (1-\eta, 1]\}$.

It is well known that a word x is Sturmian if and only if its factor complexity function $p_x(n) = n + 1$ for all n, and that for all η, χ, we have $\mathcal{L}(x_\eta) = \mathcal{L}(x_{\eta,\chi})$. As discussed above, Sturmian words are special instances of *toric words*, i.e., they each code the orbit of an irrational rotation on a torus partitioned into finitely many connected components.

A typical Sturmian word that we will consider in this thesis is the so-called order word of the powers of 2 and 3: this word $2322323223\cdots$ records the order in which the powers of 2 and 3 occur, i.e., $2, 3, 4, 8, 9, 16, 27, 32, 64, 81, \ldots$, and is a two-dimensional *cutting sequence*, as illustrated in Fig. 1.1. Indeed, the word records the order in which the diagonal line intersects horizontal and vertical grid lines.

The *gradient* γ of a cutting sequence is the gradient (rise over run) of the diagonal cutting line when the diagram is rescaled in a way that both horizontal and vertical grid lines are unit distance apart. In our example, $\gamma = \log_3 2$. It is easy to check, by studying the intersections with vertical lines, that if the cutting line starts at the origin, then the cutting sequence is $\lambda(x_\gamma)$, where x_γ is the characteristic Sturmian word with slope γ, and λ is the substitution that maps 0 (representing intersection with a vertical grid line) to 0 and 1 to 01. The word $\lambda(x_{\log_3 2})$, in turn, is the characteristic Sturmian word of slope $\log_6 2$.

Lemma 1.1.3. *A two-dimensional cutting sequence with gradient γ is a Sturmian word with slope $\frac{\gamma}{\gamma+1}$.*

Proof. We assume without losing generality that vertical grid lines are intersected more often (i.e., $\gamma < 1/2$), and that the letter 0 records such intersections. As observed above, we can associate the cutting sequence with a one-dimensional rotation: the

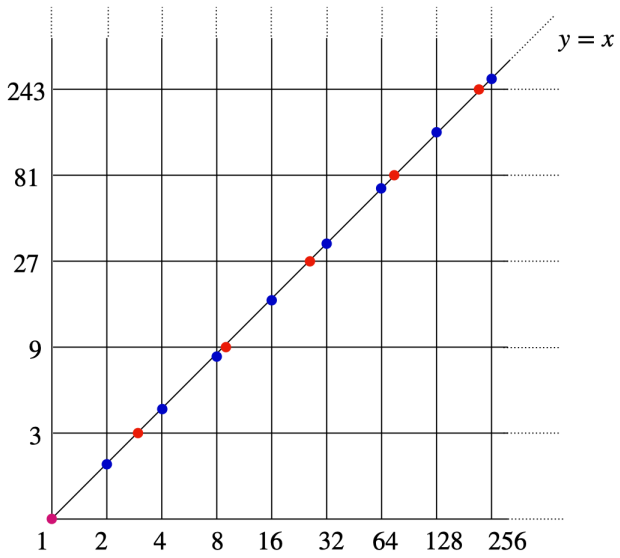

Figure 1.1: Order word as a two-dimensional cutting sequence. Cutting sequences may be generalised to higher dimensions, e.g., the word recording the order in which powers of $2, 3, 5$ occur.

arc of length $1 - \gamma$ corresponds to the word 0; depending on whether the cutting sequence starts with 0 or 1, the arc of length γ respectively corresponds to 01 (Case vertical) or 10 (Case horizontal).

The desired Sturmian system is obtained by "duplicating" the arc of length γ. We thus have three arcs: the original $I_\gamma, I_{1-\gamma}$, and the copy I'_γ. We associate the interval of length $1 + \gamma$ with the one-dimensional torus, and arrange our arcs as $I_{1-\gamma}, I'_\gamma, I_\gamma$ in Case vertical, and $I'_\gamma, I_{1-\gamma}, I_\gamma$ in Case horizontal.

The original arcs I_γ and $I_{1-\gamma}$ correspond to the letters 1 and 0 respectively, while the new copy I'_γ corresponds to the letter 0. As we rotate in discrete steps of γ, this has the effect of "delegating" the duty of I_γ to print two letters between itself and its copy.

Since the length of the inserted copy is the same as the rotation step, the order of visiting the original arcs in the new system is the same as that in the original system.

Finally, to meet the syntactic definition of a Sturmian system, we scale the circumference from $1 + \gamma$ to 1, which makes the discrete rotation step to be $\frac{\gamma}{1+\gamma}$, as desired. □

1.1.3 S-adic Words

Given an alphabet Σ, let $\mathcal{S}(\Sigma)$ denote the set of all substitutions from Σ to Σ^+. Let $S \subseteq \mathcal{S}(\Sigma)$ be a set of substitutions. We say that a word $x \in \Sigma^\omega$ is *S-adic* if there exist a sequence of words $(x^{(n)})_{n\in\mathbb{N}}$ and a sequence of substitutions $\boldsymbol{\sigma} = (\sigma_n)_{n\in\mathbb{N}}$ over S such that $x^{(0)} = x$, and for all n we have $x^{(n)} = \sigma_n\left(x^{(n+1)}\right)$. We call $\boldsymbol{\sigma}$ the *directive sequence* and say that x is directed by $\boldsymbol{\sigma}$. As we shall see, a stronger notion of direction is *generation*, whereby x is generated by $\boldsymbol{\sigma} = (\sigma_n)_n$ if there is some augmentation $\hat{\boldsymbol{\sigma}} = ((\sigma_n, a_n))_n$ such that $x = \lim_{n\to\infty} \sigma_0 \cdots \sigma_n(a_n)$.

Naturally, a substitutive word x, which is a fixed point of a non-erasing substitution σ is directed (and generated) by σ^ω. More generally, if x has an ultimately periodic directive sequence of non-erasing substitutions, then it is morphic.

A directive sequence $\boldsymbol{\sigma}$ is called *primitive* if for every n, there exists k such that for every letter a, the image $\sigma_n \cdots \sigma_{n+k}(a)$ contains all letters of the alphabet Σ. If there moreover exists a uniform k for all n, then the directive sequence is called *strongly primitive*. In this thesis, we will typically work with primitive directive sequences; we refer the reader to [10] for a comprehensive survey on the subject of S-adicity.

An observant reader might argue that every word $x = a_0a_1a_2\cdots$ is vacuously S-adic for $S = \{\sigma_1, \ldots, \sigma_{|\Sigma|}, \tau_1, \ldots, \tau_{|\Sigma|}\}$ over the alphabet $\Sigma \cup \{b\}$, where σ_i replaces the distinguished letter b with $a_i \in \Sigma$ and is the identity function elsewhere, and τ_i replaces b with ba_i and is the identity function elsewhere. We then see that x is directed (and generated) by $\sigma_{a_0}\tau_{a_1}\tau_{a_2}\cdots$. (Indeed, $\lim_{n\to\infty} \sigma_{a_0}\tau_{a_1}\cdots\tau_{a_n}(b) = x$.)

The construction above indicates that we need to exercise caution while formulating decision problems about S-adic words.

Lemma 1.1.4. *It is undecidable, given a finite set S of substitutions and an ω-regular language $L \subseteq \Sigma^\omega$, whether there exists a directive sequence $\boldsymbol{\sigma} \in S^\omega$ that generates a word x which is included in L.*

Proof. We reduce from the infinite Post Correspondence Problem (ω-PCP). An instance of this problem gives alphabets Σ, Γ (we may assume $|\Gamma| = 2$), and morphisms $\sigma_0, \sigma_1 : \Sigma \to \Gamma^+$, and asks whether there exists a word $x \in \Sigma^\omega$ such that $\sigma_0(x) = \sigma_1(x)$. This problem was shown to be undecidable with $|\Sigma| \geq 9$ in [43].

For our decision problem, we take our alphabet

$$\hat{\Sigma} = (\Sigma \times \{0,1\}) \cup (\Sigma' \times \{0,1\}) \cup \Gamma \cup \{\#, \$, \triangleright, \bot\},$$

where Σ' is a set consisting of primed copies c' for each $c \in \Sigma$.

For each letter $c \in \Sigma$, we define the substitution τ_c as $\# \mapsto \$ \cdot (c', 0)$, and $(d', b) \mapsto (c', 1-b)(d, 1-b)$, and $(d, b) \mapsto (d, 1-b)$, and the identity elsewhere. Note that

$$\tau_{c_0} \cdots \tau_{c_n}(\#) = \$ \cdot (c_0', b)(c_1, b) \cdots (c_n, b),$$

where b has the same parity as n.

We define the substitution σ as $(c', b) \mapsto \sigma_b(c)$, and $(c, b) \mapsto \sigma_b(c)$, and $\$ \mapsto \triangleright$, and $\triangleright \mapsto \bot$ and the identity elsewhere, and observe that

$$\sigma\tau_{c_0} \cdots \tau_{c_n}(\#) = \triangleright \cdot \sigma_{(n \bmod 2)}(c_0 \cdots c_n).$$

Note that the image on the right is subsequently fixed by all substitutions τ_c, and any subsequent application of the substitution σ leads to $\triangleright$ being replaced by $\bot$.

We now ask whether there exists a directive sequence over the set of these substitutions which generates some infinite word beginning with the letter $\triangleright$. By construction, if there is such a word, it must be generated by a directive sequence of the form $\sigma\tau_{c_0}\tau_{c_1} \cdots$, each composition applied to the letter $\#$. By definition and the continuity of σ_0, σ_1, this is equivalent to

$$\sigma_0(c_0 \cdots c_n \cdots) = \lim_{n\to\infty} \sigma_0(c_0 \cdots c_n) = \lim_{n\to\infty} \sigma_1(c_0 \cdots c_n) = \sigma_1(c_0 \cdots c_n \cdots),$$

i.e., the existence of a solution to the given instance of the undecidable ω-PCP. □

The vacuous construction notwithstanding, the point of directive sequences is to provide a perspective that conveys more structural information about the word beyond a simple renaming of letters.

The most relevant example is the class of Sturmian words, where the directive sequence of a characteristic Sturmian word is readily derived from the continuous fraction expansion of the slope η, and similarly the directive sequence of a general Sturmian word is obtained from the continuous fraction expansion of η and the so-called Ostrowski expansion of the intercept χ in base η. We refer to [11, Prop. 2.7, also see remark after Thm. 2.10] for the following facts; see also [38] for a detailed exposition of the S-adic nature of Sturmian words and Arnoux-Rauzy words.

We now give a brief technical introduction of the number-theoretic terminology above, see also [19]. For an irrational number $\eta \in (0, 1)$, its *continued fraction expansion* is the unique sequence $(a_n)_{n\geq 1}$ of positive integers such that

$$\eta = \lim_{n\to\infty} \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_n}}}} = \lim_{n\to\infty} \frac{p_n}{q_n}. \tag{1.6}$$

We denote the above as $\eta = [0; a_1, a_2, \ldots]$.

The series of fractions p_n/q_n are known as the *convergents* to η. The numerators and denominators satisfy the recurrences $p_0 = 0$, $q_0 = 1$, $p_1 = 1$, $q_1 = a_1$, and $(p_{n+2}, q_{n+2}) = a_{n+2} \cdot (p_{n+1}, q_{n+1}) + (p_n, q_n)$ for all $n \geq 0$. The convergents are the *locally best approximants* of η: for every $n \in \mathbb{N}$, $p \in \mathbb{Z}$, and $0 < q < q_n$, we have that $|q_n\eta - p_n| < \min_{p\in\mathbb{N}} |q\eta - p|$, which implies that $\left|\eta - \frac{p_n}{q_n}\right| < \min_{p\in\mathbb{N}} \left|\eta - \frac{p}{q}\right|$.

For example, the inverse of the golden ratio $1/\phi = 0.618\ldots$ has the continued fraction expansion $[0; 1, 1, \ldots]$, and $1/\phi^2$ has the continued fraction expansion $[0; 2, 1, 1, \ldots]$.

The *Ostrowski numeration system in base* η is based on [1] the sequence of q_n for natural numbers, and $\theta_n = q_n\eta - p_n$ for reals in $[-\eta, 1-\eta]$. For any $\chi \in [-\eta, 1-\eta]$, there exists a sequence $(b_n)_{n\geq 1}$ over $\mathbb{N}$ such that (i) $0 \leq b_1 < a_1$, (ii) $0 \leq b_n \leq a_n$ for all $n \geq 2$, (iii) for all n, we have $b_{n+1} = a_{n+1}$ only if $b_n = 0$, and

$$\chi = \sum_{n=1}^{\infty} b_n \theta_{n-1}. \tag{1.7}$$

We refer to $(b_n)_{n\geq 1}$ as an Ostrowski expansion of χ in base η. Conversely, every $(b_n)_{n\in\mathbb{N}}$ satisfying (i-iii) is an Ostrowski expansion of some χ in base η, i.e., the infinite sum converges to a value in $[-\eta, 1-\eta]$.[2] It is known [45, Fact 2.8] that for an integer $m > 0$, we have $m = \sum_{i=0}^{\infty} b_i q_{i-1}$, where the real number $\chi = m\eta - n \in [-\eta, 1-\eta)$ is given by $\sum_{i=0}^{\infty} b_i\theta_{i-1}$ with only finitely many $b_i \neq 0$.

If $\chi \notin \mathbb{Z} + \eta\mathbb{Z}$ or if $\chi \in \mathbb{Z} + \eta\mathbb{N}$, then χ has a unique Ostrowski expansion in base η. Otherwise, χ has two expansions in base η: one with $b_{2n+1} = a_{2n+1}$ for all large enough n, which we call the upper Ostrowski expansion, and the other with $b_{2n} = a_{2n}$, which we call the lower Ostrowski expansion.

We are ready to describe the S-adic nature of Sturmian words. The set S of *Sturmian substitutions* is $\{\lambda_0, \lambda_1, \rho_0, \rho_1\}$. Here λ_i "inserts i to the left", i.e., $\lambda_0(0) = 0$, $\lambda_0(1) = 01$, $\lambda_1(0) = 10$, and $\lambda_1(1) = 1$, and similarly, ρ_i "inserts i to the right". A convenient observation is that λ_i and ρ_i commute, e.g., $\lambda_0 \circ \rho_0$ is the same substitution as $\rho_0 \circ \lambda_0$.

Let $x_{\eta,\chi}$ be a Sturmian word with slope $\eta = [0; a_1 + 1, a_2, \ldots]$ and intercept χ. Then there exists (by [11, Prop. 2.8][3]) a unique Ostrowski expansion $(b_n)_{n\in\mathbb{N}}$ of χ in

[1] The numeration system can also be defined in terms of $|\theta_n|$; in this paper we follow [11, Sec. 2.4], [19], and [45] and work with the θ_n-based system.

[2] To check this, observe that θ_n alternates between positive and negative; $\theta_0 = \eta$, $\theta_1 = a_1\eta - 1$; $\theta_{n+2} = a_{n+2}\theta_{n+1} + \theta_n$.

[3] Caution: The proof of the cited proposition leaves the case of $\chi \in \mathbb{Z} + \mathbb{Z}_{\leq -1}\eta$ incomplete. We discuss a patch in Lem. 1.1.6.

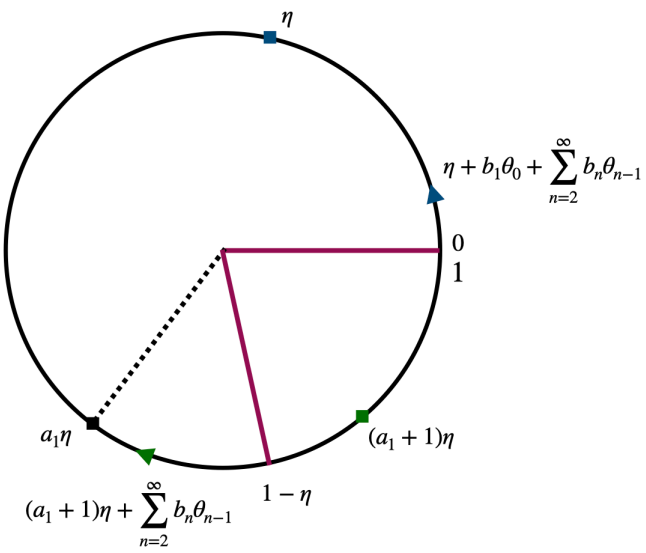

Figure 1.2: Sturmian word with slope $\eta = [0; a_1+1, a_2, \ldots]$ (here $a_1 = 3$) and intercept $\chi = \sum_{n=1}^{\infty} b_n\theta_{n-1}$, visualised as a rotation. We print the letter 1 if we land in the minor arc subtended by the magenta radii, and 0 otherwise.

base η (also known as the *formal intercept*) such that x is directed by the sequence $(\tau_n)_{n\in\mathbb{N}}$ where

$$\tau_n = \lambda_0^{a_{2n+1}-b_{2n+1}} \rho_0^{b_{2n+1}} \lambda_1^{a_{2n+2}-b_{2n+2}} \rho_1^{b_{2n+2}}. \tag{1.8}$$

Conversely, the rules of Ostrowski expansion guarantee that each $(\tau_n)_{n\in\mathbb{N}}$ obtained from the expansions $(a_n)_{n\in\mathbb{N}}$ and $(b_n)_{n\in\mathbb{N}}$ as above is weakly primitive and directs a Sturmian word. (In fact, in the sense of Def. 1.1.15, this directive sequence will *generate* a unique Sturmian word.) Observe that we can indeed unpack every τ_n to obtain a bona fide directive sequence over $S = \{\lambda_0, \lambda_1, \rho_0, \rho_1\}$.

Observe that for characteristic Sturmian words, $\chi = 0$, and hence $b_n = 0$ for all n. Thus, characteristic Sturmian words have directive sequences over $\{\lambda_0, \lambda_1\}$. Indeed, the following lemma illuminates the self-similar nature of Sturmian words.

Lemma 1.1.5. *Let x be a Sturmian word with slope $\eta = [0; a_1 + 1, a_2, a_3, \ldots]$, and intercept χ whose base-η Ostrowski expansion is $(b_1, b_2, b_3, \ldots)$. We have that $x = \lambda_0^{a_1-b_1} \circ \rho_0^{b_1}(x')$, where x' is a Sturmian word (with letters renamed) having slope $\eta' = [0; a_2 + 1, a_3, a_4, \ldots]$, and intercept χ' whose base-η' Ostrowski expansion is $(b_2, b_3, b_4, \ldots)$.*

Proof. The proof is a technically involved extension of the ideas used in Lem. 1.1.3.

We interpret the Sturmian word x as the coding of a rotation on the one-dimensional torus, as illustrated in Fig. 1.2. The key observation is that between any two occurrences of the letter 1, there are either a_1 or a_1+1 occurrences of the letter 0. In other words, there is a word x' such that $x = \tau(x')$, where the substitution τ maps 0 to 0 and 1 to $0^{a_1-b}10^b$ for some b, i.e., $\tau = \lambda_0^{a_1-b} \circ \rho_0^b$. We shall prove that $b = b_1$, and that x' is Sturmian with the desired properties.

Observe that we can deduce the number of intervening occurrences of 0 by inspecting whether the orbit lands in $[a_1\eta, 1-\eta)$ immediately before it lands in $[1-\eta, 1)$. If it does, then we have a_1+1 intervening occurrences of 0, otherwise there are only a_1.

We define a new dynamical system which records when the orbit falls in the interval $[1-a_1\eta, 1)$. The length of this interval is $\eta + 1 - (a_1+1)\eta = \theta_0 + |\theta_1|$, the latter being in terms of the base-η numeration system. Observe that if the current point is in the length-θ_0 subinterval, then the next point will be $|\theta_1|$ clockwise; otherwise the next point will be θ_0 counter-clockwise. Thus, the dynamics can equivalently be given as a one-dimensional rotation. We associate the circle with $(0, 1]$, rotate by $\eta' = |\theta_1|/(\theta_0 + |\theta_1|)$, and partition into arcs $(0, 1-\eta']$ and $(1-\eta', 1]$ that respectively correspond to letters 1 and 0. It is now clear that this dynamical system, of which we shall define x' to be a coding, is a Sturmian system.

Let us show, through routine algebraic manipulation, that the slope η' has continued fraction expansion $[0; a_2+1, a_3, \ldots]$. Recall that $\eta' = \frac{1-(a_1+1)\eta}{1-a_1\eta}$. We begin with $\eta = [0; a_1+1, a_2, a_3, \ldots]$. Inverting, we get $1/\eta = [a_1+1; a_2, a_3, \ldots]$. Subtracting a_1+1 from both sides, we get $\frac{1-(a_1+1)\eta}{\eta} = [0; a_2, a_3, \ldots]$. Inverting again, we get $\frac{\eta}{1-(a_1+1)\eta} = [a_2; a_3, a_4, \ldots]$. Adding 1 to both sides, we get $\frac{1-a_1\eta}{1-(a_1+1)\eta} = [a_2+1; a_3, a_4, \ldots]$. Finally, inverting gives us the desired equality.

Next, let us define the initial point $\eta' + \chi'$ of the new system in terms of the initial point $\eta + \chi$ of the original rotational system. Recall that $\chi = \sum_{n=1}^{\infty} b_n\theta_{n-1}$, where the digits b_n follow the rules of the Ostrowski enumeration system. We set $\chi' = \frac{-1}{1-a_1\eta}\sum_{n=2}^{\infty} b_n\theta_{n-1}$. As illustrated in Fig. 1.2, the starting point $\eta' + \chi'$ of the new system corresponds to the point $(a_1+1)\eta + \sum_{n=2}^{\infty} b_n\theta_{n-1}$ of the original system.

We verify that this map is well-defined. Since we identify the torus with $[0, 1)$, our convention for disambiguating Ostrowski expansions is as follows: there are infinitely many n for which $b_{2n+1} < a_{2n+1}$. We use Θ as shorthand for $\sum_{n=2}^{\infty} b_n\theta_{n-1}$. We check that $\Theta < |\theta_1|$; if $b_1 = 0$ then $\Theta \geq -\theta_0$, otherwise $\Theta \geq -\theta_0 + |\theta_1|$.

Indeed, to maximise Θ, we choose $b_{2n+1} = a_{2n+1}$. By the recurrence relation on the sequence on θ_n, we get $\sum_{n=1}^{\infty} a_{2n+1}\theta_{2n} = (\theta_3 - \theta_1) + (\theta_5 - \theta_3) + \cdots = |\theta_1|$. Similarly, to

minimise Θ, we choose $b_{2n} = a_{2n}$. If $b_1 = 0$, this yields a minimum of $-\theta_0$; otherwise the minimum is $(a_2 - 1)\theta_1 - \theta_2 = -\theta_0 + |\theta_1|$.

We now argue that the coding τ from x' to x ought to map 1 to $0^{a_1 - b_1}10^{b_1}$. For this, observe that by construction, the number occurrences of the letter 0 "skipped over" when we map the initial point in the original system to the initial point in the new system is equal to $a_1 - b_1$: using Θ as shorthand for $\sum_{n=2}^{\infty} b_n\theta_{n-1}$, these skipped occurrences correspond to landing points from the starting point $\eta + b_1\eta + \Theta$ through to $a_1\eta + \Theta$.

Finally, it remains to establish that χ' has base-η' Ostrowski expansion $(b_2, b_3, \ldots)$. Recall (since $\theta_0 = \eta$) that $\chi' = \frac{-1}{1-a_1\theta_0}\sum_{n=2}^{\infty} b_n\theta_{n-1}$. It would suffice for us to prove that for all n, $\theta'_n = \frac{-\theta_{n+1}}{1-a_1\theta_0}$.

We do so by induction. The inductive step is immediate, since already we have $a'_n = a_{n+1}$. Hence, $\theta'_{n+2} = a'_{n+2}\theta'_{n+1} + \theta'_n = a_{n+3}\frac{-\theta_{n+2}}{1-a_1\theta_0} + \frac{-\theta_{n+1}}{1-a_1\theta_0} = \frac{-\theta_{n+3}}{1-a_1\theta_0}$. One of the base cases is also trivial, since by construction we have $\eta' = \theta'_0 = \frac{-\theta_1}{1-a_1\theta_0}$. We only need to prove, via algebraic manipulations, that $\theta'_1 = \frac{-\theta_2}{1-a_1\theta_0}$.

We have that $\theta'_1 = (a_2 + 1)\theta'_0 - 1$. Upon simplifying and taking the common denominator $1 - a_1\theta_0$ for the entire expression, the numerator becomes $-a_2\theta_1 - \theta_1 - 1 + a_1\theta_0$. Since $\theta_1 = a_1\theta_0 + \theta_0 - 1$, we can replace $(-\theta_1 - 1 + a_1\theta_0)$ with $-\theta_0$. Our numerator then becomes $-a_2\theta_1 - \theta_0$, which is simply $-\theta_2$, as desired.

This concludes the proof of all the properties asserted in the statement. □

We make a (relatively) brief comment on the distinction between lower and upper mechanical words when the intercept $\chi \in \mathbb{Z} + \mathbb{Z}_{\leq -1}\eta$: we describe which formal intercept applies. See [19, Sec. 4] (and also the construction of [11, Prop. 2.7]) for more technical details.

Lemma 1.1.6. *Let x be a Sturmian word with slope $\eta = [0; 1 + a_1, a_2, a_3, \ldots]$ and intercept $\chi \in \mathbb{Z} - m\eta$ for some $m \geq 1$. We choose the formal intercept with $b_{2n} = a_{2n}$ for all but finitely many n if and only if x is a lower mechanical word.*

Proof Sketch. The principle is as follows. We have the choice to converge to χ as either $(q_{2l} - m)\eta$ from above, or $(q_{2l-1} - m)\eta$ from below. After m steps, the convergents must agree with the partition $\chi + m\eta$ falls into. Thus, if we have a lower mechanical word, we must converge from above, and if we have an upper mechanical word, we must converge from below. The rest of the proof sketches how the inductive construction of the previous lemma mirrors this intuition.

As a sanity check, observe that this holds when $m = 1$: when we have a lower mechanical word, the intercept is in $[-\eta, 1 - \eta)$, and we need $\chi = -\eta$ to be equal to

the largest-magnitude negative sum $\sum_i a_{2i}\theta_{2i-1}$; in the other case, we need $\chi = 1-\eta$ to be equal to the largest possible positive sum. We henceforth assume that $m \geq 2$.

Choose l minimally so that $m \leq q_l$. (This choice of l is at least 1 because $m \geq 2$.) Express $m' = q_l - m$ in the base-η Ostrowski numeration system as $b_1 q_0 + b_2 q_1 + \ldots + b_l q_{l-1}$. We have that $q_l - q_{l-1} > m' \geq 0$. By the defining recurrence $q_0 = 1$, $q_1 = a_1 + 1$, and $q_{n+2} = a_{n+2}q_{n+1} + q_n$, we have that:

- It cannot be that $b_l = a_l$. Otherwise, $m' + q_{l-1}$ would be at least $q_l + (q_{l-1} - q_{l-2})$, a contradiction. (The case $l = 1$ can be checked manually.)
- If $b_l = a_l - 1$ and $l \geq 2$, it must be that $b_{l-1} = 0$. Otherwise, by similar reasoning, $m' + q_{l-1}$ would be at least q_l.

Let us now evaluate the number $\chi \in [\eta, 1-\eta]$ that $-m\theta_0$ is equivalent to modulo 1. By [45, Fact 2.8], we can equivalently consider $b_1\theta_0 + \ldots + b_{l-1}\theta_{l-2} + b_l\theta_{l-1} - \theta_l$.

We now have the options to expand $-\theta_l$ as either

$$\sum_{i=1}^{\infty} a_{l+2i}\theta_{l+2i-1}$$

or

$$\theta_{l-1} - \theta_l - \theta_{l-1} = \theta_{l-1} + (a_{l+1} - 1)\theta_l + \sum_{i=2}^{\infty} a_{l+2i-1}\theta_{l+2i-2}.$$

The correct option for the Sturmian word x at hand is induced by the iterative construction we provided in Lem. 1.1.5. We begin at Step 1; when x is a lower mechanical word, the torus is oriented as $[0,1)$ at odd steps and as $(0,1]$ at even steps. (The case of upper mechanical words is vice versa.) At step i, we choose b_i so that the next starting point falls within the next one-sided open interval. We observe that our options diverge at Step l, where the slope is $\eta_{\ell-1}$

At this step, moving the starting point by $b_l \cdot \eta_{l-1}$ as usual to get the new starting point lands on an endpoint of the new Step-$(l+1)$ system corresponding to the intercept $-\eta_l$. The case distinction is made based on whether we landed on the closed or open endpoint. Let us assume we work with lower mechanical words; the upper case is analogous with parities flipped.

When l is even, $l+1$ is odd, and our usual construction lands on the closed endpoint. We thus use the first option, whence $b_{2n} = a_{2n}$ for all but finitely many n. When l is odd, $l+1$ is even, and our usual construction lands on the open endpoint, and must be altered according to the second option. Even in this case $b_{2n} = a_{2n}$ for all but finitely many n. This argument, when followed in (cumbersome) detail, would prove the lemma. □

A generalisation of Sturmian words of an especially S-adic nature is the class of *Arnoux-Rauzy words*. Consider the alphabet $\{0, 1, \ldots, d-1\}$, and defined the Arnoux-Rauzy substitutions $\lambda_0, \ldots, \lambda_{d-1}, \rho_1, \ldots, \rho_{d-1}$ analogously to the Sturmian substitutions, i.e., $\lambda_a(a) = a$ and $\lambda_a(b) = ab$ if $a \neq b$, and dually, $\rho_a(a) = a$ and $\rho_a(b) = ba$ if $b \neq a$. A word x is Arnoux-Rauzy if it is directed by a weakly primitive sequence over the set of Arnoux-Rauzy substitutions. In such a sequence, for every a, either λ_a or ρ_a occurs infinitely often. The set of 2-letter Arnoux-Rauzy words is precisely that of Sturmian words.

The most common example of an Arnoux-Rauzy word is the *Tribonacci word* $0102010\cdots$, which is the fixed point of the morphism which maps 0 to 01, 1 to 02 and 2 to 0. The Tribonacci word is directed by the sequence $(\lambda_0\lambda_1\lambda_2)^\omega$.

For any Arnoux-Rauzy word x over a d-letter alphabet, the factor complexity function $p_x(n) = (d-1)n + 1$. However, when $d > 2$, a word with this factor complexity is not necessarily an Arnoux-Rauzy word.

1.1.4 Recurrent Words

A factor u of an infinite word x is said to be *recurrent* if there are infinitely many occurrences of u in x. We define $\mathcal{L}^{\mathsf{rec}}(x)$ to be the set of recurrent factors of x, and declare $\mathcal{L}^{\mathsf{rec}}_\ell(x)$ as shorthand for $\mathcal{L}^{\mathsf{rec}}(x) \cap \Sigma^\ell$. The word x is recurrent if all its factors are recurrent, i.e., $\mathcal{L}(x) = \mathcal{L}^{\mathsf{rec}}(x)$. The Fibonacci word and Thue-Morse word are examples of recurrent words.

Lemma 1.1.7. *A word $x \in \Sigma^\omega$ is recurrent if and only if there exists an initial word $u_0 \in \Sigma^+$, and a sequence of words $(u_n)_{n\in\mathbb{N}}$ defined using words $v_0, v_1, \ldots \in \Sigma^*$ as $u_{n+1} = u_n v_n u_n$, such that $x = \lim_{n\to\infty} u_n$.*

A word $x \in \Sigma^\omega$ is called *disjunctive* if $\mathcal{L}(x) = \mathcal{L}^{\mathsf{rec}}(x) = \Sigma^*$. If, furthermore, the word x admits factor frequencies and for every $u \in \Sigma^*$, we have that $f_x(u) = 1/|\Sigma|^{|u|}$, then we say that x is a *normal* word. The *Champernowne word* $0123456789101112\cdots$, which is formed by concatenating the decimal representations of the positive integers, is a quintessential example of a normal word.

Let u be a recurrent factor of a word x. A *return word* to u in x is a finite word r such that the word ru is a factor of x that has exactly two occurrences of u: once as a prefix, and once as a suffix. The set of return words to u in x is denoted $\mathcal{R}_x(u)$. For example, the return words to 010 in the Fibonacci word are $01, 010$.

A *recurrent return word* to u in x is a return word $r \in \mathcal{R}_x(u)$ such that ru is a recurrent factor of x. The set of recurrent return words to u in x is denoted $\mathcal{R}^{\mathsf{rec}}_x(u)$.

Lemma 1.1.8. *Given any recurrent factor u, a word x can be factorised as the concatenation $vr_1r_2\cdots$, where $r_i \in \mathcal{R}_x(u)$ for each $i \geq 1$. Moreover, if x is recurrent, then v is a proper suffix of some $r_0 \in \mathcal{R}_x(u)$.*

Proof. We shall first observe that if x begins with an occurrence of u, then it can be factorised purely as a concatenation of return words to u. Consider the shortest prefix w that contains the first two occurrences of u: it is of the form r_0u, where r_0 is a return word, say of length ℓ_0. We repeat the process on $x(\ell_0, \infty)$, and continue ad infinitum to obtain the desired factorisation.

The above construction can be used to obtain the factorisation $vr_1r_2\cdots$, where the beginning of r_i coincides with the i-th occurrence of u. It remains to prove that the prefix v of x is a suffix of some return word r_0 to u in the case that x is recurrent.

Indeed, we observe that by our factorisation process, vu is a factor of x with exactly one occurrence of u. Consider the second occurrence of vu; let it be at index $i > 0$. This causes an occurrence of u in x at index $i + |v|$, which is to the right of the one at index $|v|$. In particular, let the occurrence of u in x that immediately precedes the one at index $i + |v|$ be at index $j < i$. Now, $x(j, i + |vu|)$ is a word of the form ru, where r is a return word that has v as its suffix. □

A recurrent word x is called *uniformly recurrent* if for every factor u, the set $\mathcal{R}_x(u)$ of its return words is finite. Equivalently, for every factor u, there is a bound on the index of its first occurrence, and on the gaps between consecutive occurrences of u. We can thus define a *return-time function* $R_x : \mathbb{N} \to \mathbb{N}$ as one that satisfies

$$R_x(\ell) \geq \max_{u \in \mathcal{L}_\ell(x)} \max_{r \in \mathcal{R}_x(u)} |r|, \tag{1.9}$$

i.e., R_x upper bounds the length of the longest return word to some length-ℓ factor. Note that we can assume R_x is increasing: indeed, if u is a prefix of u' then any return word r to u will be a prefix of some return word r' to u'.

If $R_x(\ell) \in \mathcal{O}(\ell)$, then we say that x is *linearly recurrent*. For example, the Fibonacci word and Thue-Morse word are linearly recurrent.

An immediate consequence of Lem. 1.1.8 is captured in the following definition.

Definition 1.1.9. *If a word x is uniformly recurrent, then for any word u of length ℓ, the index of the first occurrence of u is at most $R_x(\ell) - 1$. Thus, if x is effective, then a return-time function R_x is computable if and only if $\mathcal{L}(x)$ is recursive. Should one (and hence both) of these conditions hold, we say that x is* effectively uniformly recurrent.

We enumerate $\mathcal{L}_\ell(x)$ by recording the length-ℓ factors occurring at any set of $R_x(\ell)$ consecutive indices.

We note that we sometimes work with *almost-periodic* words, which constitute a slightly more general class than uniformly recurrent words, and were studied by Semënov [58, 68]. A word x is almost-periodic if for every recurrent factor u, the set $\mathcal{R}_x(u)$ of its return words is finite, i.e., there is a bound on the gaps between consecutive occurrences of u.

For almost-periodic words x, we define R_x similarly as we did for uniformly recurrent words, except that in (1.9), we let the range of u be recurrent factors, and the range of r be recurrent return words. It is often technically convenient to also declare a *window function* $W_x : \mathbb{N} \to \mathbb{N}$ as $W_x(\ell) = R_x(\ell) + \ell$. We observe, by a straightforward induction that

$$W_x^n(\ell) = \ell + \sum_{i=0}^{n-1} R_x \circ W_x^i(\ell), \tag{1.10}$$

where W_x^n denotes the n-fold composition of the function W_x.

Analogously to the case where x is uniformly recurrent, we would like to enumerate $\mathcal{L}_\ell^{\mathsf{rec}}(x)$ by recording the length-ℓ factors at $R_x(\ell)$ consecutive indices, even when x is almost-periodic. For this, we rely on the following structure of almost-periodic words.

Lemma 1.1.10. *Given any recurrent factor u, an almost-periodic word x can be factorised as the concatenation $vr_1r_2\cdots$, where $r_i \in \mathcal{R}_x^{\mathsf{rec}}(u)$ for each $i \geq 1$, and moreover, all factors of $r_1r_2\cdots$ that have length at most $|u|$ are recurrent.*

Proof. Consider the factorisation $v'r'_1r'_2\cdots$ of x given by Lem. 1.1.8. For each j, let k_j be the starting index of r'_j. Observe that by construction, these indices are precisely the indices at which the factor u occurs in x. In particular, a factor ru where $r \in \mathcal{R}_x(u)$ can only occur at index k_j for some j.

Thus, if $r \in \mathcal{R}_x(u)$ is not a recurrent return word, then $r = r'_j$ for only finitely many j. Since x is almost-periodic, the set $\mathcal{R}_x(u)$ is finite, and thus for all but finitely many j, we have $r'_j \in \mathcal{R}_x^{\mathsf{rec}}(u)$. We choose m large enough to also ensure that the suffix does not contain any non-recurrent factors of length $|u|$, and set $r_i = r'_{i+m}$, and $v = v'r'_1 \dots r'_m$. □

The above lemma suggests that almost-periodic words are technically not very different from uniform recurrent words. In order to formalise this intuition, we define a function $V_x : \mathbb{N} \to \mathbb{N}$ to be a *volatility function* for x and R_x if it has the following properties.

1. If $m \geq V_x(\ell)$, and a factor u of length ℓ occurs in x at index m, then there is another occurrence of u in x at index at most $m + R_x(\ell)$.

2. If $m \geq V_x(\ell) + R_x(\ell)$, and a factor u of length ℓ occurs in x at index m, then there is another occurrence of u in x at index at least $m - R_x(\ell)$.

These requirements allow us to assume that V_x is increasing, and together ensure that if $m \geq V_x(\ell)$, then a factor u of length ℓ is recurrent in x if and only if u occurs at one of the indices $m, m+1, \ldots, m+R_x(\ell)$. Naturally, if x is uniformly recurrent, then the function which is identically 0 is a volatility function.

An obvious implementation of $V_x(\ell)$ is obtained through Lem. 1.1.10 as follows. Construct the prescribed factorisation for all recurrent factors u of length at most ℓ, and take the length of the longest prefix v obtained in the process.

Lemma 1.1.11. *Let x be an almost-periodic word. If x is effective, then $\mathcal{L}(x)$ is recursive if and only if we can compute a window function W_x and a volatility function V_x. Should one (and hence both) of these conditions hold, we say that x is* effectively almost-periodic.

Proof. We record that a lemma of this nature can also be found in [60, Lem. 4.2.5]. Note that the "if" direction follows immediately by definition; we shall focus on proving the converse.

We first give a subroutine to determine whether a given factor u is recurrent in x, by showing that the sets of recurrent factors, and non-recurrent factors are both recursively enumerable. Observe that if a factor u is not recurrent in an almost-periodic word x, it has a final occurrence in x: thus, there exists a length ℓ, computable through brute enumeration, such that there is no length-ℓ factor of x has u as its suffix. The index of the last occurrence of u can be deduced by inspecting the longest factor that has u as its suffix. Similarly, if u is recurrent in x, then there exists a length ℓ such that all length-ℓ factors of x contain an occurrence of u.

One similarly computes the length of the longest recurrent return word by enumerating recurrent factors that have exactly two occurrences of u: this proves that W_x is computable. To compute (the Lem. 1.1.10-based implementation of) V_x, we use the subroutine to compute the index of the last occurrence of a non-recurrent factor: we give to it as input words of the form ru, where r is not a recurrent return word, and non-recurrent words u' that are not longer than u. □

1.1.5 Effective Uniform Recurrence

In this subsection, we shall discuss the uniform recurrence of the classes of words we have introduced thus far.

As we shall see throughout this thesis, uniform recurrence is closely related to topological compactness. The following standard result sees the familiar Euclidean topology at play.

Proposition 1.1.12. *Sturmian words admit factor frequencies and are uniformly recurrent.*

Proof. Let x be a Sturmian word with slope η and intercept χ such that $x(n) = 0$ if $\chi + n\eta \in \mathbb{Z} + [0, 1-\eta)$, and $x(n) = 1$ if $\chi + n\eta \in \mathbb{Z} + [1-\eta, 1)$. We view x as coding the orbit of a discrete rotation by η on the torus identified by the interval $I_\varepsilon = [0, 1)$, and partitioned into intervals $I_0 = [0, 1-\eta)$ and $I_1 = [1-\eta, 1)$.

For each word $u \in \{0,1\}^*$, we inductively define intervals I_u: given I_u and a letter a, we define I_{au} to be $I_a \cap (I_u - \eta)$, where the latter is the interval I_u rotated through by $-\eta$. We inductively observe that I_u, if non-empty, is such that its left end is closed, and the right end is open.

Observe that there is an occurrence of the factor u at index n in x if and only if $\chi + n\eta$ modulo 1 falls in the interval I_u. By the Kronecker-Weyl equidistribution theorem, this happens with frequency equal to the length of I_u: we have thus established that x admits factor frequencies.

To prove uniform recurrence, we will prove that given a recurrent factor u (i.e., I_u is non-empty), there is a uniform bound R such that for any ζ, its orbit falls in I_u within R steps. Indeed, let $J_u(0)$ be an open subset of I_u. For $n \in \mathbb{N}$, we define $J_u(n) = J_u(0) - n\eta$, i.e., $J_u(n)$ is the set of points whose orbit falls in $J_u(0)$ at the n-th step. By the equidistribution theorem, the orbit of every point on the torus must visit $J_u(0)$: thus, $J_u(0), J_u(1), \ldots$ constitute an open cover of the compact one-dimensional torus. This open cover necessarily admits a finite subcover $J_u(0), \ldots, J_u(R)$. In other words, we have proven what we desired at the beginning of this paragraph. □

Lemma 1.1.13. *If a Sturmian word x is effective, then it is effectively uniformly recurrent.*

Proof. Uniform recurrence follows from Prop. 1.1.12. Recall (Lem. 1.1.1) that since x is effective, having a recursive factor language as desired is equivalent to the factor-complexity function being computable. But we already know that $p_x(n) = n + 1$ for Sturmian words. □

In the rest of this subsection, we shall discuss how (effective) uniform recurrence is established through symbolic means. For instance, consider the following standard fact, which gives yet another way of observing that Sturmian words are uniformly recurrent, and also establishes the property for Arnoux-Rauzy words.

Lemma 1.1.14. *If a word x has a primitive directive sequence $\boldsymbol{\sigma} = (\sigma_n)_{n\in\mathbb{N}}$, then it is uniformly recurrent.*

Proof. We first gather some technical tools that will aid the proof.

Recall that by definition, there exists a sequence $x^{(0)}, x^{(1)}, \ldots$ of words such that $x^{(0)} = x$, and $x^{(n)} = \sigma_n\left(x^{(n+1)}\right)$ for all n. Define a sequence $a_0, a_1, \ldots$ of letters such that $a_n = x^{(n+1)}(0)$ for all n. By construction, we have the property that the word $\sigma_0(a_0)$ begins with $x(0)$, and for all $n \geq 1$, the word $\sigma_n(a_n)$ begins with a_{n-1}.

The above property will prove particularly convenient beyond this lemma too, so we shall formalise it in a definition.

Definition 1.1.15. *Let $\boldsymbol{\sigma} = (\sigma_n)_{n\in\mathbb{N}}$ be a directive sequence over a set S of substitutions on the alphabet Σ. An augmented directive sequence $\hat{\boldsymbol{\sigma}} = ((\sigma_n, a_n))_{n\in\mathbb{N}}$ is called* congenial *if for all $n \geq 1$, the word $\sigma_n(a_n)$ is non-empty and begins with a_{n-1}.*

The convenience of congenial sequences stems from the following property, which can be proved via a straightforward induction.

Lemma 1.1.16. *Let $\hat{\boldsymbol{\sigma}} = ((\sigma_n, a_n))_{n\in\mathbb{N}}$ be congenial, and for each $n \geq 1$ we denote $\sigma_n(a_n)$ by $a_{n-1}v_n$. We have that for all $n \geq 1$,*

$$\sigma_0 \cdots \sigma_n(a_n) = \sigma_0(a_0) \cdot \sigma_0(v_1) \cdot \sigma_0\sigma_1(v_2) \cdots (\sigma_0 \cdots \sigma_{n-1}(v_n)).$$

In our current context, the directive sequence $\boldsymbol{\sigma}$ being primitive guarantees that each σ_n is non-erasing, and moreover that there are infinitely many n such that v_n is non-empty. Consequently, $\lim_{n\to\infty} \sigma_0 \cdots \sigma_n(a_n) = x$. We say that x is *congenially generated* by $\boldsymbol{\sigma}$, or more specifically, by $\hat{\boldsymbol{\sigma}}$.

We are now ready to prove the uniform recurrence of x, i.e., every factor u of x is recurrent, and the gaps between its consecutive occurrences are bounded. Consider a factor u of x, and let n be the smallest number for which u is a factor of the prefix $\sigma_0 \cdots \sigma_n(a_n)$. Let r be the smallest number such that for all $a \in \Sigma$, the letter a_n occurs in the word $\sigma_{n+1} \cdots \sigma_{n+r}(a)$: this r exists because $\boldsymbol{\sigma}$ is primitive. Let $R = \max_a |\sigma_0 \cdots \sigma_{n+r}(a)|$.

It is now easy to check that for any word $y \in \Sigma^\omega$, the word u is a factor of any length-$(2R)$ factor of $\sigma_0 \cdots \sigma_{n+r}(y)$. In particular, we can choose y to be $x^{(n+r+1)}$,

which proves that u is a recurrent factor of x, the gaps between whose consecutive occurrences are bounded. □

We can, of course, prove stronger properties if we assume more structure on the directive sequence, e.g., in the case that x is the fixed point of a primitive substitution σ. It is folklore that such words x are, in fact, *linearly* recurrent and admit factor frequencies. The effectiveness of these properties is implicit in classical textbooks; we discuss it here for completeness.

Lemma 1.1.17. *Primitive substitutive words are effectively linearly recurrent and admit computable factor frequencies.*

Proof. In the proof, we shall assume a fixed order on the alphabet Σ, and associate with the substitution σ an adjacency matrix M, whose (j, i)-th entry records the number of times the j-th letter a_j appears in the image $\sigma(a_i)$ of the i-th letter a_i. Because σ is primitive, we can assume, by replacing σ with some σ^k, that M has strictly positive entries, which allows us to use Perron–Frobenius theory.

We first show that x is effectively linearly recurrent, i.e., $R_x(n) \leq Ln$ for some computable L. The proof of linear recurrence in [29, Prop. 25] shows that $L = C \cdot \max_{a\in\Sigma} |\sigma(a)| \cdot R_x(2)$, where C is such that for all $n \geq 1$, $\max_{a\in\Sigma} |\sigma^n(a)| \leq C \cdot \min_{a\in\Sigma} |\sigma^n(a)|$.

It suffices to take $C = (\max_{i,j}(v_i/v_j))^2$, where $(v_i)_{i=1}^{|\Sigma|}$ are the strictly positive entries of a left eigenvector $v^\top$ of M corresponding to the Perron-Frobenius eigenvalue ρ (i.e., the positive one with maximum absolute value). Indeed, for the i-th letter a, $|\sigma^n(a)| = \sum_j (M^n)_{j,i}$. The expression on the right is the sum of entries in the i-th column of M^n, and is lower bounded by $\sum_j (v_j/\max_j v_j) \cdot (M^n)_{j,i}$, and upper bounded by $\sum_j (v_j/\min_j v_j) \cdot (M^n)_{j,i}$. These bounds respectively evaluate to the i-th entries of $(v^\top M^n)/(\max_j v_j)$ and $(v^\top M^n)/(\min_j v_j)$. Since $v^\top$ is an eigenvector, we get that

$$\rho^n (\min_j v_j)/(\max_j v_j) \leq |\sigma^n(a)| \leq \rho^n (\max_j v_j)/(\min_j v_j).$$

We now show how to compute (an upper bound on) $R_x(2)$. First, we can compute $\mathcal{L}_x(2)$ by starting with $x(0, 2)$, and recording the length-2 factors that are produced upon repeatedly applying σ. The set of factors will saturate within $|\Sigma|^2$ iterations. In particular, we can find k large enough such that for all $a \in \Sigma$, $\sigma^k(a)$ contains every factor in $\mathcal{L}_x(2)$. Clearly, $R_x(2) \leq 2 \max_a |\sigma^k(a)|$. This proves effective linear recurrence.

We note that the textbook discussions in [66, Chap. 5.4] and [34, Chap. 3.8.5] prove x has computable factor frequencies. We present the techniques here for clarity

in exposition. Recall the adjacency matrix M of σ. It follows that the frequency of the i-th letter a in x is the i-th entry of the eigenvector corresponding to the Frobenius eigenvalue.

This same idea is extrapolated to describe factor frequencies as follows. We define an alphabet Σ_n, whose letters correspond to words in $\mathcal{L}_n(x)$ (this alphabet can indeed be enumerated as x is effectively linearly recurrent), and the word $x_n \in \Sigma_n^\omega$, where $x_n(i)$ corresponds to the factor $x(i, i+n)$. We define the substitution σ_n such that $\sigma_n(u)$ is the ordered list of the first $|\sigma(u(0))|$ length-n factors of $\sigma(u)$. Crucially, this substitution on Σ_n is primitive [66, Lem. 5.3], and we can again use Perron-Frobenius theory to obtain the frequency of factor $u \in \Sigma^n$, which is a letter in Σ_n. □

Note, however, that arbitrary substitutive words are not necessarily recurrent or almost-periodic, nor must they admit factor frequencies. Indeed, consider the word $2110^6 1^{18} 0^{54} \cdots$, which is the fixed point of the substitution that maps 2 to 211, 1 to 000, and 0 to 111. The frequency of the letter 0 in increasingly longer prefixes both drops below $1/3$ and exceeds $2/3$ infinitely often.

It is also routine to show that recurrence properties are preserved under applying substitutions.

Lemma 1.1.18. *Let $\sigma : \Sigma \to \Gamma^*$ be a substitution, and let $x \in \Sigma^\omega$ be such that $\sigma(x)$ is an infinite word. If x has one of the following properties, then so does $\sigma(x)$.*

1. *recurrent*
2. *(effectively) uniformly recurrent*
3. *(effectively) linearly recurrent*
4. *(effectively) almost-periodic*

Proof. To formalise the proof, consider the *attribution function* α from indices of $\sigma(x)$ to indices of x. We define $\alpha(j)$ to be the smallest index i such that $j < |\sigma(x(0, i+1))|$, i.e., the letter at index j of $\sigma(x)$ is attributed to the image of $x(i)$. We define $\alpha(J) = \bigcup_{j \in J} \alpha(j)$. In particular, when v is a factor of $\sigma(x)$ at index j, then we set $J = [j, j + |v|)$, obtain $[i, i')$ as the smallest interval that contains $\alpha(J)$, and say that the above occurrence of v is attributed to an occurrence of $u = x(i, i')$ in x at index i.

Clearly, if an occurrence of v is attributed to an occurrence of u, then the recurrence of u implies that of v. This proves item (1).

In case x is almost-periodic, we can decide whether a given word v is in $\mathcal{L}(\sigma(x))$. Indeed, an occurrence of v will always be attributed to a factor u that is at most $(R_x(1) + V_x(1)) \cdot |v|$ in length. This is because the gaps between occurrences of non-erasing letters (letters a for which $\sigma(a)$ is not the empty word) are eventually bounded by $R_x(1)$, and are at most $V_x(1)$. For the remainder of the proof, we declare U_x as shorthand for $R_x + V_x$.

The gaps between the occurrences of u are themselves bounded by $U_x(U_x(1) \cdot |v|)$. Finally, the gaps between the occurrences of v must be bounded by $(\max_{a \in \Sigma} |\sigma(a)|) \cdot U_x(U_x(1) \cdot |v|)$. This proves items (2), (4), and also item (3): if $R_x(n)$ is linear in n, then the bound on the gaps between occurrences of v will also be linear in $|v|$. □

An immediate consequence of Lem. 1.1.17 and Lem. 1.1.18 is that primitive morphic words are effectively linearly recurrent. In fact, we have the following stronger result of Durand [33, Thm. 1, Thm. 3, Sec. 4, Sec. 5] (see also [32, Lem. 4] and the statement of [36, Thm. 24(1)]).

Proposition 1.1.19 (Durand)**.** *A morphic word x is uniformly recurrent if and only if it is primitive morphic. Given substitutions σ, τ such that the morphic word x is the image under τ of the fixed point of σ, we can decide whether x is uniformly recurrent. If it is uniformly recurrent, it is in fact effectively linearly recurrent, and we can further compute substitutions σ', τ' such that σ' is primitive, τ' is letter-to-letter, and x is the image under τ' of the fixed point of σ'.*

We shall frequently use the above proposition to obtain more amenable presentations of morphic words. (We remark that this extends the discussion preceding Lem. 1.1.2.)

1.2 Automata

In this thesis, automata will serve as a means to recognise (ω-)regular languages, and also be viewed as an interesting model of computation in their own right.

A *(deterministic) automaton* $\mathcal{A}$ is given by $(Q, q_{\mathsf{init}}, \Sigma, \delta)$, where Q is a finite nonempty set of states, q_{init} is the initial state, Σ is the alphabet it runs over, and $\delta : Q \times \Sigma \to Q$ is the transition function which prescribes the next state, given the current state and the letter being read. We naturally extend the domain of δ to $Q \times \Sigma^*$ by defining $\delta(q, \varepsilon) = q$ for all q, and $\delta(q, ua) = \delta(\delta(q, u), a)$. Recall that a language $L \subseteq \Sigma^*$ is *regular* if and only if there exists an automaton $(Q, q_{\mathsf{init}}, \Sigma, \delta)$ and a set of states $F \subseteq Q$ such that $u \in L$ if and only if $\delta(q_{\mathsf{init}}, u) \in F$.

For a word $x = a_0a_1a_2\cdots$ over the alphabet Σ, the *run* $\mathcal{A}(x)$ of $\mathcal{A}$ on x is the word $(q_0, a_0)(q_1, a_1)(q_2, a_2)\cdots$ over the alphabet $Q \times \Sigma$, where $q_0 = q_{\mathsf{init}}$, and for all n we have $\delta(q_n, a_n) = q_{n+1}$. We similarly define "the run v of $\mathcal{A}$ on u, starting in state q" in the obvious way. This is useful when u is a factor of x, and we are interested in the fragment of the run produced while reading u.

Automata may also be *nondeterministic*: in this case, q_{init} is replaced by a set Q_{init} of initial states, and δ is replaced by a transition relation $\Delta \subseteq Q \times \Sigma \times Q$. For a word $x = a_0a_1a_2\cdots$ over the alphabet Q, a word $(q_0, a_0)(q_1, a_1)(q_2, a_2)\cdots$ is a run of $\mathcal{A}$ on x if $q_0 \in Q_{\mathsf{init}}$, and for all n we have that $(q_n, a_n, q_{n+1}) \in \Delta$. We will work with nondeterministic automata only occasionally; throughout this thesis, "an automaton" means "a deterministic automaton" unless specified otherwise.

An automaton $\mathcal{A}_0 = (Q, q_{\mathsf{init}}, \Sigma, \delta)$ may be extended into a *transducer* $\mathcal{A}$ by specifying an output alphabet Γ and an output function $\tau : Q \times \Sigma \to \Gamma^*$. The transducer, upon reading a letter a in the state q, prints $\tau(q, a)$. The output $\mathcal{A}(x)$ of the transducer on input x is then $\tau(\mathcal{A}_0(x))$, i.e., we view τ as a substitution from $(Q \times \Sigma)$ to Γ^*. If this is a non-erasing substitution, we say that $\mathcal{A}$ is a *non-erasing* transducer.

1.2.1 Monadic Second-Order Logic

Monadic second-order logic (MSO) is an extension of first-order logic that allows quantification over subsets of the universe. Such subsets can be viewed as unary (i.e., monadic) predicates. We will only be interpreting MSO formulae over expansions of the structure $\langle \mathbb{N}; < \rangle$ with unary predicates.

For example, $\langle \mathbb{N}; <, \mathsf{Primes} \rangle$ is an expansion with the unary predicate Primes, where $\mathsf{Primes}(n)$ (alternatively written as $n \in \mathsf{Primes}$) is true if and only if n is prime. Other examples of unary predicates include Pow_2, i.e., the set of powers of 2, $\mathsf{Squares}$, and P_{TM}, defined as $n \in P_{\mathsf{TM}}$ if and only if $x_{\mathsf{TM}}(n) = 1$, where x_{TM} is the Thue-Morse word $01101001\cdots$.

Given a structure $\langle \mathbb{N}; <, P_0, \ldots, P_{d-1} \rangle$, we define its *characteristic word* x as an infinite word over the alphabet $\{0, 1, \ldots, 2^d - 1\}$, where each letter is given by a d-bit binary string, and the 0-th bit is least significant. The i-th bit of $x(n)$ is 1 if and only if $n \in P_i$. For example, the characteristic word of $\langle \mathbb{N}; <, \mathsf{Pow}_2, \mathsf{Pow}_3 \rangle$ is $0312100012\cdots1\cdots2\cdots1\cdots$. Naturally, the characteristic word of $\langle \mathbb{N}; <, P_{\mathsf{TM}} \rangle$ is x_{TM}.

MSO formulae use first-order variables $t_1, t_2, t_3, \ldots$ for numbers, and second-order variables $T_1, T_2, T_3, \ldots$ for sets. An MSO formula φ is constructed using the grammar

$$\varphi := t \in P \mid t \in T \mid t < t \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists t.\ \varphi \mid \exists T.\ \varphi. \tag{1.11}$$

Of course, we can define disjunction and indeed all other logical connectives using negation and conjunction, and we can define universal quantification using negation and existential quantification.

We can express $t_1 = t_2$ as $\neg(t_1 < t_2) \wedge \neg(t_2 < t_1)$; we can express $t = 0$ as $\forall t_1.\ t \leq t_1$; and we can can express $t_1 = t_0 + 1$ as $t_0 < t_1 \wedge \forall t.\ (t_0 < t) \Rightarrow (t_1 \leq t)$. We can then define T to be the set of numbers congruent to 1 modulo 4 as

$$\varphi_{1 \bmod 4}(T) := (1 \in T) \wedge (\{0, 2, 3\} \cap T = \{\}) \wedge \forall t.\ (t \in T \Leftrightarrow t + 4 \in T).$$

Formulae where all variables are bound to a quantifier are known as *sentences*. For example, the sentence

$$\exists T.(\varphi_{1 \bmod 4}(T) \wedge \forall t_1 \exists t_2.\ (t_1 < t_2) \wedge t_2 \in T \wedge t_2 \in \mathsf{Primes})$$

states that there are infinitely many primes congruent to 1 modulo 4, which is a consequence of a classical result of Dirichlet.

The *MSO theory* of a structure is the set of all sentences that are true for that structure, i.e., sentences that evaluate to true when the constant predicate symbols $P_0, \ldots, P_{d-1}$ are interpreted as prescribed by the structure. For example, the above sentence is in the MSO theory of $\langle \mathbb{N}; <, \mathsf{Primes} \rangle$. A structure has a *decidable* MSO theory if this set is recursive. If the theory of $\langle \mathbb{N}; <, \mathsf{Primes} \rangle$ were shown to be decidable, we would, at least in principle, be able to resolve the Twin Prime Conjecture.

Büchi's foundational work [18] described the set of structures in which a sentence holds true, in terms of characteristic words over the appropriate alphabet $\Sigma = \{0, 1, \ldots, 2^d - 1\}$. The set of satisfying characteristic words is an *ω-regular language* $L \subseteq \Sigma^\omega$, i.e., a language of the form $\bigcup_{i=1}^{d} U_i V_i^\omega$, where $U_1, V_1, \ldots, U_d, V_d$ are regular languages, and $V_1, \ldots, V_d$ do not contain the empty word. Equivalently, one can effectively construct a nondeterministic automaton $\mathcal{A} = (Q, Q_{\mathsf{init}}, \Sigma, \Delta)$ and give a set of states $F \subseteq Q$ such that $x \in L$ if and only if there is a run of $\mathcal{A}$ on x that visits some state in F infinitely often. We say that L is recognised by the nondeterministic Büchi automaton $\mathcal{A}$.

It is possible, and often desirable, to recognise ω-regular languages using deterministic automata. We use (deterministic) parity automata for this purpose. We

equip an automaton $(Q, q_{\mathsf{init}}, \Sigma, \delta)$ with a colouring function $\mathsf{col} : Q \to \mathbb{N}$, and declare that $\mathcal{A}$ accepts x if and only if the maximum colour seen infinitely often along the run $\mathcal{A}(x)$ is even.

Nondeterministic Büchi automata and deterministic parity automata recognise precisely the class of ω-regular languages, and one can effectively convert between the two, see e.g., [64]. Consequently, we have the following.

Theorem 1.2.1. *Consider the structure $\langle \mathbb{N}; <, P_0, \ldots, P_{d-1} \rangle$, and let x be its characteristic word. The structure has a decidable MSO theory if and only if given any deterministic parity automaton $\mathcal{A}$, we can decide whether $\mathcal{A}$ accepts x.*

We introduce the following notation for brevity. Let x be an effective word. We use Acc_x to denote the decision problem which takes a deterministic parity automaton $\mathcal{A}$ as input, and asks whether $\mathcal{A}$ accepts x. One of the central results of this thesis is showing the decidability of Acc_x, where x is the characteristic word of the structure $\langle \mathbb{N}; <, \mathsf{Pow}_2, \mathsf{Pow}_3 \rangle$. (More generally, we can take the sets of powers of any two multiplicatively independent integers.)

We shall often use the following simple observation.

Lemma 1.2.2. *Let $\mathcal{A} = (Q, q_{\mathsf{init}}, \Sigma, \delta_Q, \Gamma, \tau)$ be a transducer, and let x, y be infinite words such that $y = \mathcal{A}(x)$. We have that Acc_y reduces to Acc_x.*

Proof. We reduce the query of whether an automaton $\mathcal{B} = (R, r_{\mathsf{init}}, \Gamma, \delta_R, \mathsf{col}_R)$ accepts y to the query of whether an automaton

$$\mathcal{C} = (R \times \{0, \ldots, \max_r \mathsf{col}_R(r)\} \times Q, (r_{\mathsf{init}}, 0, q_{\mathsf{init}}), \Sigma, \delta, \mathsf{col})$$

accepts x. The automaton $\mathcal{C}$ produces y as the output of $\mathcal{A}(x)$, and further simulates the run of $\mathcal{B}$ on y.

Define $\delta((r, i, q), a) = (\delta_R(r, \tau(q, a)), j, \delta_Q(q, a))$, where the colour j is the maximum colour of a state in the run of $\mathcal{B}$ on $\tau(q, a)$ starting in state r. In other words, if $\tau(q, a) = b_0 \cdots b_{\ell-1}$, and the run fragment is $(r_0, b_0) \cdots (r_{\ell-1}, b_{\ell-1})$ where $r_0 = r$, then $j = \max\{\mathsf{col}_R(r_0), \ldots, \mathsf{col}_R(r_{\ell-1})\}$.

Finally, we define $\mathsf{col}((r, i, q)) = i$ for all states of $\mathcal{C}$, and thus observe that by construction, the highest colour seen infinitely often in the run of $\mathcal{B}$ on y is the same as the highest colour seen infinitely often in the run of $\mathcal{C}$ on x. The reduction is thus complete. □

Semënov gave the following characterisation (see e.g., [22, Thm. 5]) of words x for which Acc_x is decidable.

Theorem 1.2.3. *Let $x \in \Sigma^\omega$. We have that Acc_x is decidable if and only if the following two problems, which take as input a regular language $L \subseteq \Sigma^+$, are decidable.*

1. *Is $x \in L\Sigma^\omega$?*
2. *Is $x \in (\Sigma^* L)^\omega$?*

We draw inspiration from this characterisation in defining a recurrent word $x \in \Sigma^\omega$ to be *effectively recurrent* if given any regular language $L \subseteq \Sigma^+$, we can decide whether x has a prefix in L, and whether $L \cap \mathcal{L}(x)$ is nonempty. Observe that by definition, for a recurrent word x, we have that Acc_x is decidable if and only if x is effectively recurrent.

There are examples of recurrent words x such that x is effective, $\mathcal{L}(x)$ is recursive, but Acc_x is undecidable (see Prop. 2.2.5). We shall later see in Prop. 2.3.2 that it suffices to only require that we can decide, given any regular language L, whether $L \cap \mathcal{L}(x)$ is nonempty. We also subsequently discuss the proof of Thm. 1.2.3 itself. (We note that Prop. 2.2.5 is readily evident from an alternate formulation of Semënov's characterisation established in [8, Thm. 3]. Nevertheless, our exposition allows us to prove a "prefix-independent" version of Semënov's result in Prop. 2.3.3.)

We record the following immediate addendum to Lem. 1.1.18.

Lemma 1.2.4. *Let x be an effectively recurrent word, and let σ be a substitution such that the word $y = \sigma(x)$ is infinite. We have that y is an effectively recurrent word.*

Disjunctive words are a class of recurrent words whose interaction with automata can be described particularly well. Recall that any automaton induces a (directed) graph, where the vertices correspond to states, and the edges a derived from transitions, i.e., there is an edge from q to q' if and only if $\delta(q, a) = q'$ for some letter a. (We allow edges from a vertex to itself.) A *strongly connected component* of a graph is a maximal set of vertices such that there is a path between any pair of vertices. By definition, connected components partition the set of states, and form a partial order: we say that R is below R' if there is a path from R' to R, but not vice versa. A strongly connected component is a *bottom strongly connected component* if there is no outgoing edge, i.e., an edge from a vertex in the component to a vertex outside it.

Lemma 1.2.5. *For any automaton $\mathcal{A}$ and disjunctive word x, the set of states visited infinitely often in the run of $\mathcal{A}$ on x is a bottom strongly connected component of the graph induced by $\mathcal{A}$. Thus, if a disjunctive word x is effective, then Acc_x is decidable, i.e., x is effectively recurrent.*

Proof. The second part of the statement indeed follows readily from the first: we keep feeding the letters of x as input to the given automaton until it enters a bottom strongly connected component. We then check whether the highest colour of a state in this component is even.

We therefore focus on proving that the set of states visited infinitely often is a bottom strongly connected component. We proceed in two steps: (i) for every strongly connected component R that has an outgoing edge, there is a word u_R such that for all $q \in R$, we have $\delta(q, u_R) \notin R$; (ii) for every bottom strongly connected component R, and every state $q \in R$, there exists a word u_q such that for every state $q' \in R$, the run of $\mathcal{A}$ on u_q starting in q' visits q. Since all words are recurrent factors of x, this would prove that $\mathcal{A}$ eventually enters a bottom strongly connected component, and having done so, visits all its states infinitely often.

In the following arguments, we shall assume that the states $q_1, \ldots, q_k$ of a strongly connected component R are ordered. We only prove step (ii) in an inductive manner. The proof of step (i) is obtained by simply replacing "visits q" with "visits a state not in R".

We define u_1 to be the shortest word such that the run on u_1 starting in q_1 visits q, and maintain the invariant that for all $q' \in \{q_1, \ldots, q_i\}$, the run on u_i starting in q' visits q. For the inductive step, we take $u_{i+1} = u_i v$, where v is the shortest word such that the run on $u_i v$ starting in q_{i+1} visits q. Finally, we take $u_q = u_k$. □

1.2.2 Equivalence with Semigroups

Regular languages, of both finite and infinite words, can be viewed from an algebraic perspective. Recall that a *group* G is a set equipped with an operation $\cdot$ that combines two elements to produce a third element within the set. (When the context is clear, we omit the operation sign $\cdot$ and simply juxtapose the elements being combined, as we do when denoting multiplication or word concatenation.)

This operation must: (i) be associative, i.e., $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$, (ii) admit an identity element e (also called the neutral element; note that it will necessarily be unique), i.e., for all elements g, we have $e \cdot g = g \cdot e = g$, and (iii) be such that every element has an inverse, i.e., for every g_1, there exists an element g_2 such that $g_1 \cdot g_2 = g_2 \cdot g_1 = e$. For example, the nonzero real numbers with the multiplication operation form a group.

If we drop the third requirement of every element needing to have an inverse, then our set with its binary operation is called a *monoid*. For example, the real numbers with multiplication form a monoid, but not a group. The set Σ^* with the operation

of word concatenation forms a *non-commutative* monoid, i.e., it is not necessary that $uv = vu$.

If we also drop the second requirement on the existence of the identity element, then our set with its binary operation is called a *semigroup*. For example Σ^+, along with the operation of word concatenation, is a semigroup but not a monoid.

Any semigroup M can be augmented into a monoid by simply adjoining a fresh element e and declaring $e \cdot e = e$, $e \cdot m = m \cdot e = m$ for all $m \in M$, and keeping all other operation results unchanged.

A *morphism* h from Σ^* into a monoid M is a function that maps the empty word to the neutral element, and satisfies $h(uv) = h(u) \cdot h(v)$ for all u, v. It is well known that a language $L \subseteq \Sigma^*$ is regular if and only if there exists a finite monoid M, a subset $F \subseteq M$, and a morphism $h : \Sigma^* \to M$ such that $u \in L$ if and only if $h(u) \in F$. We say that the monoid M recognises L.

The above fact is readily seen by defining the *transition monoid* $M_{\mathcal{A}}$ of an automaton $\mathcal{A} = (Q, q_{\mathsf{init}}, \Sigma, \delta)$. Let M be the finite monoid of functions from Q to Q, with function composition being the binary operation. Observe that each letter $a \in \Sigma$, when input to δ, defines a function $\delta_a : Q \to Q$. The monoid $M_{\mathcal{A}}$ is simply the submonoid generated by the set of δ_a.

We now recall that analogously, ω-semigroups (see [63, Sec. 7]) are an equivalent way to recognise ω-regular languages. Formally, *an ω-semigroup* $M = (M_f, M_\omega)$ is a "two-sorted" algebra equipped with the following operations:

1. A binary operation defined on M_f and denoted multiplicatively,
2. A mapping $M_f \times M_\omega \to M_\omega$, called the *mixed product*, also denoted multiplicatively,
3. An infinite product π that maps infinite sequences over M_f to an element of M_ω.

These operations must satisfy the following associativity properties:

1. M_f, equipped with the binary operation, is a semigroup.
2. For every $m_1, m_2 \in M_f$ and $m_3 \in M_\omega$, we have that $(m_1 m_2)m_3 = m_1(m_2 m_3)$.
3. For every sequence $(m_n)_{n\in\mathbb{N}}$ over M_f, and every strictly increasing sequence $(k_n)_{n\in\mathbb{N}}$ of indices, we have that $\pi(m_0, m_1, \ldots) = \pi(m_0 \cdots m_{k_0}, m_{k_0+1} \cdots m_{k_1}, \ldots)$.

4. For every $m \in M_f$ and every sequence $(m_n)_{n\in\mathbb{N}}$ over M_f, we have that $m \cdot \pi(m_0, m_1, \ldots) = \pi(m, m_0, m_1, \cdots)$.

The last two conditions show that the infinite product $\pi(m_0, m_1, \ldots)$ can also be denoted multiplicatively as $m_0 m_1 \cdots$. Indeed, an ω-semigroup can be intuited as a semigroup where infinite products are defined. An immediate example is $\Sigma^+ \cup \Sigma^\omega$ with word concatenation, where $M_f = \Sigma^+$, and $M_\omega = \Sigma^\omega$.

Given ω-semigroups M_1, M_2 a morphism h of ω-semigroups is a pair h_f, h_ω such that h_f is a semigroup morphism from $M_{1,f}$ into $M_{2,f}$, and h_ω is a map from $M_{1,\omega}$ to $M_{2,\omega}$ preserving the mixed product and infinite product, i.e., for every $m_1 \in M_{1,f}$ and $m_2 \in M_{1,\omega}$, $h_f(m_1)h_\omega(m_2) = h_\omega(m_1 m_2)$, and for every sequence $(m_n)_{n\in\mathbb{N}}$ over $M_{1,f}$, we have $h_f(m_0)h_f(m_1)\cdots = h_\omega(m_0 m_1 \cdots)$. We shall thus omit subscripts and denote the application of a morphism by simply h. As an immediate example, a non-erasing substitution σ defines a morphism from $\Sigma^+ \cup \Sigma^\omega$ to $\Sigma^+ \cup \Sigma^\omega$.

A set $L \subseteq \Sigma^+ \cup \Sigma^\omega$ is recognised by a morphism h from Σ^∞ into an ω-semigroup M if there exists a subset $H \subseteq M_f \cup M_\omega$ such that $L = h^{-1}(H)$. A language $L \subseteq \Sigma^\omega$ is ω-regular if and only if it is recognisable by a finite ω-semigroup [63, Thm. 7.6]. See [63, Sections 8,9] respectively for effective translations from Büchi automata to ω-semigroups, and vice versa. Furthermore, analogous to the finite-word case, there is a notion of the (finite) syntactic ω-semigroup of an ω-regular language [63, Sec. 11].

A few remarks addressing presentation concerns are in order. A result of Wilke [71] showed that for finite ω-semigroups M, the infinite product is fully determined by the function $M_f \to M_\omega$ that maps m to m^ω [63, Thm. 7.1]. This combinatorial result is proven through Ramsey's theorem, and gives a method to construct ω-semigroups through Wilke algebras. We refer the reader to [63, Sec. 7] for details of how the semigroup structure of M_f determines an extension into an ω-semigroup through *linked pairs*.

We use the equivalence of finite ω-semigroups and automata through the following lemma.

Lemma 1.2.6. *Let $M = (M_f, M_\omega)$ be a finite ω-semigroup, and let $x \in M_f \cup M_\omega$. Extend M_f into a monoid M'_f by adjoining a distinguished neutral element 1_M: for all $m \in M'_f$, $1_M \cdot m = m \cdot 1_M = m$. We can construct an automaton $\mathcal{A}_x$ over the alphabet M'_f that accepts $m_0 m_1 \cdots \in (M'_f)^\omega$ if and only if the infinite product $m_0 m_1 \cdots$ (defined as the possibly finite product of the subsequence $m_{i_0} m_{i_1} \cdots$ obtained by discarding all 1_M terms) equals x.*

The constructed automaton can either be nondeterministic with Büchi acceptance, or deterministic with parity acceptance, as is convenient.

For example, consider the ω-regular language $(a^*b)^\omega$ over the alphabet $\{a, b\}$, i.e., the language of infinite words containing infinitely many occurrences of the letter b. We recognise it algebraically as follows. Let $M_f = \{e, m\}$, where $e \cdot e = e, e \cdot m = m \cdot e = m \cdot m = m$. Let $M_\omega = \{\hat{e}, \hat{m}\}$, where $e^\omega = \hat{e}$ and $m^\omega = \hat{m}$. We define the morphism h to map a to e, and b to m. Observe that a word x is in the language if and only if $h(x) = \hat{m}$.

1.2.3 The Krohn–Rhodes Framework

To prove results of the form, "if a word x has a property P, then the run $\mathcal{A}(x)$ also has property P", it is especially helpful to express $\mathcal{A}$ as the composition of several simpler automata, and prove the preservation of property P for the runs of each of the simpler automata. This is where we use Krohn–Rhodes theory.

We first describe the nature of the "simpler" automata. Recall that for an automaton $\mathcal{A} = (Q, q_{\mathsf{init}}, \Sigma, \delta)$, each letter a defines a function $\delta_a : Q \to Q$, and the monoid generated by composing these functions is the transition monoid of $\mathcal{A}$. An automaton $\mathcal{A}$ is called a *reset automaton* if for every letter a, the function δ_a is either constant or the identity function. An automaton $\mathcal{A}$ is called a *permutation automaton* or a *group automaton* if for every letter a, the function δ_a is a permutation of Q, or in other words, the transition monoid of $\mathcal{A}$ is in fact a group. We say that $\mathcal{A}$ is a *permutation-reset automaton* if for each letter a, the function δ_a is either constant, or a permutation of Q.

We also mention the special class of *counter-free automata*. The automaton $\mathcal{A}$ is said to be *counter-free* if there do not exist a subset Q' of states and a word $u \in \Sigma^+$ such that δ_u induces a non-trivial permutation of the states of Q'. Equivalently, the transition monoid of $\mathcal{A}$ has no non-trivial subgroups. We remark that counter-free automata are of particular interest because of their equivalence with first-order-definable languages (also known as aperiodic languages), as established by the McNaughton–Papert theorem [55].

We are now ready to state the Krohn–Rhodes theorem. This theorem was originally proven [49], and is often stated, for semigroups; we refer to [56] for its formulation in terms of automata.

Theorem 1.2.7 (Krohn–Rhodes)**.** *Let $\mathcal{A}$ be an automaton. We can compute automata $\mathcal{B}_1, \ldots, \mathcal{B}_k$, and a substitution σ such that for any word x, we have that*

$$\mathcal{A}(x) = \sigma \circ \mathcal{B}_k \circ \cdots \circ \mathcal{B}_1(x),$$

and moreover:

1. *each automaton $\mathcal{B}_i$ is either a permutation automaton or a (two-state) reset automaton,*
2. *if $\mathcal{B}_i$ is a permutation automaton then its transition group is homomorphic to a subgroup of the transition monoid of $\mathcal{A}$. In particular, if $\mathcal{A}$ is counter-free, then each automaton $\mathcal{B}_i$ is a reset automaton.*

Indeed, as [54, Introduction] notes, the subtlety lies in requirement (2): a particular technical difficulty is to ensure that we do not introduce spurious permutations while decomposing a counter-free automaton. For most of the purposes in this thesis, however, we will only rely on the decomposition into permutation-reset automata. This part is considerably easier to prove, and we give a proof below.[4]

Proposition 1.2.8 (Lightweight Krohn–Rhodes)**.** *Let $\mathcal{A} = (Q, q_{\mathsf{init}}, \Sigma, \delta)$ be an automaton. We can compute permutation-reset automata $\mathcal{B}_1, \ldots, \mathcal{B}_k$, and a substitution σ such that for any word x, we have that*

$$\mathcal{A}(x) = \sigma \circ \mathcal{B}_k \circ \cdots \circ \mathcal{B}_1(x).$$

Proof. Our proof proceeds by induction on the number of states of $\mathcal{A}$. For the base case, we observe by exhaustive case analysis that any automaton with two or fewer states is a permutation-reset automaton. For the inductive step, we will show that any automaton $\mathcal{A}$ with n states, if not already a permutation-reset automaton, can be expressed as $\sigma' \circ \mathcal{A}' \circ \mathcal{B}$, where $\mathcal{B}$ is a permutation-reset automaton, the automaton $\mathcal{A}'$ has $n-1$ states, and σ' is a substitution.

Let us fix an ordering $q_1, \ldots, q_n$ of the states of $\mathcal{A}$, and assume $q_{\mathsf{init}} = q_1$. We will construct the run of $\mathcal{A}$ by using the permutation-reset automaton $\mathcal{B}$ to track which state the automaton $\mathcal{A}$ *cannot* be in, and using the $(n-1)$-state automaton $\mathcal{A}'$ to determine which of the remaining possible $n-1$ states $\mathcal{A}$ is actually in. We denote

[4] The proof here is inspired by the one presented on the YouTube channel of Thomas Kern, who himself attributes it to Eilenberg. It is also not too difficult to further decompose permutation-reset automata into permutation automata and two-state reset automata. This extra step, however, is counterproductive for most applications in this thesis.

the states of $\mathcal{B}$ by $r_1, \ldots, r_n$, where r_i is to be read as "the i-th state q_i is ruled out"; analogously, we denote the states of $\mathcal{A}'$ by $p_1, \ldots, p_{n-1}$, where p_i is to be read as "the automaton $\mathcal{A}$ is in the i-th of the remaining possible states". For example, if $\mathcal{B}$ is in r_2 and $\mathcal{A}'$ is in p_3, then $\mathcal{A}$ is in q_4; if $\mathcal{B}$ is in r_5 and $\mathcal{A}'$ is in p_5, then $\mathcal{A}$ is in q_6. It is now routine to describe the substitution σ'.

Formally, define the function $\mathsf{encode} : Q \times R \to P$ to determine the state of $\mathcal{A}'$ given those of $\mathcal{A}$ and $\mathcal{B}$, and define the function $\mathsf{decode} : P \times R \to Q$ to determine the state of $\mathcal{A}$ given those of $\mathcal{A}'$ and $\mathcal{B}$. We have $\sigma'((p, r, a)) = (\mathsf{decode}(p, r), a)$.

We choose $r_{\mathsf{init}} = r_2$, and thus $p_{\mathsf{init}} = p_1$.

Let us first construct $\mathcal{B} = (R, r_2, \Sigma, \delta^R)$ to be a permutation-reset automaton, given that $\mathcal{A}$ is not one already. There will thus be a letter a such that δ_a is not a permutation, and a state q_{i_a} such that for all q, we have $\delta_a(q) \neq q_{i_a}$. For such letters a, we define $\delta^R(r, a) = r_{i_a}$ for all r. In other words, the i_a-th state is ruled out immediately upon reading a. For letters a such that δ_a is a permutation, if $\delta(q_i, a) = q_j$, we also define $\delta^R(r_i, a) = r_j$. This is because if the i-th state was ruled out before permuting as prescribed by a, then the j-th state will be ruled out after. Indeed, $\mathcal{B}$ is a permutation-reset automaton.

We now complete the inductive step by constructing $\mathcal{A}' = (P, p_1, R \times \Sigma, \delta^P)$. One checks that it suffices to define

$$\delta^P(p, (r, a)) = \mathsf{encode}(\delta^Q(\mathsf{decode}(p, r), a), \delta^R(r, a)).$$

To complete the proof, we simply start with $\mathcal{A}_0 = \mathcal{A}$, our given automaton, and while $\mathcal{A}_i$ is not a permutation-reset automaton, we decompose it as $\sigma_{i+1} \circ \mathcal{A}_{i+1} \circ \mathcal{B}_{i+1}$. Since $\mathcal{A}_{i+1}$ has fewer states, this process will terminate, resulting in

$$\sigma_1 \circ \cdots \circ \sigma_{k-1} \circ \mathcal{A}_{k-1} \circ \mathcal{B}_{k-1} \circ \cdots \circ \mathcal{B}_1,$$

as desired. □

1.3 Shift Spaces

We shall use ergodic theory to study which words admit (computable) factor frequencies, and whether this property is preserved by transduction. Shift spaces will serve as our underlying dynamical systems in order to do so.

The *shift operator* T maps a word $x \in \Sigma^\omega$ to $x(1, \infty)$, i.e., it deletes the starting letter. We write $T^n x$ for the n-fold application of T to $x \in \Sigma^\omega$. Recall that the product topology that we have endowed Σ^ω with makes it a compact metric space.

A *shift system* (also called a *shift* or a *shift space*) is the dynamical system (X, T) where $X \subseteq \Sigma^\omega$ is closed and satisfies $TX \subseteq X$. When the dynamics is clear from the context, we also refer to X as a shift. Of course, Σ^ω is also a shift, and we refer to it as the *full shift* over the alphabet Σ.

1.3.1 Factor Languages

Given a word x, the *shift generated by* x is the the topological closure of the orbit $\{T^n x \mid n \in \mathbb{N}\}$ of x under T. For a shift X, we define $\mathcal{L}(X) = \{u \mid u \in \mathcal{L}(x'),\ x' \in X\}$, and as before, we declare $\mathcal{L}_n(X)$ to be shorthand for $\mathcal{L}(X) \cap \Sigma^n$. We have that if X is the shift generated by x, then $\mathcal{L}(X) = \mathcal{L}(x)$, and $X = \{x' \mid \mathcal{L}(x') \subseteq \mathcal{L}(x)\}$ [66, Prop. 4.6]. We refer to words $x' \in X$ such that $\mathcal{L}(x') = \mathcal{L}(X)$ as *typical words*.

A shift X is said to be *minimal* if it does not have a non-empty proper closed subset Y such that $TY \subseteq Y$. Therefore, the shift generated by any x' in a minimal shift X must be X itself. A word x generates a minimal shift if and only if it is uniformly recurrent [66, Prop. 4.7], and in this case we have that $X = \{x' \mid \mathcal{L}(x') = \mathcal{L}(x)\}$, i.e., all words of a minimal shift are typical.

For example, a *Sturmian shift* X_η is defined by the slope η, and contains precisely all the Sturmian words with slope η. The factor language $\mathcal{L}(X_\eta)$ is the set of all words u such that the interval I_u (as defined in Prop. 1.1.12) has nonzero length.

Shift spaces are often characterised by their *forbidden factors*, i.e., the set $\Sigma^+ \setminus \mathcal{L}(X)$. This set is upward-closed, i.e., if u is a factor of v, and u is a forbidden factor, then so is v. If the set of forbidden factors of X has a finite basis, then X is called a *shift of finite type (SFT)*. Without loss of generality, we may assume that all words in this basis have the same length $\ell \geq 2$. For example, the well known *golden shift* is the SFT formed by words in $\{0, 1\}^\omega$ where the factor 11 does not occur. This shift derives its name from the fact that the number of permissible length-n factors is precisely the $(n + 2)$-nd Fibonacci number.

More generally, if the factor language of X is regular, then we say that X is a *sofic* shift. An example is the shift with factor language $0^*(1(00)^*)^*(\varepsilon + 10^*)$, i.e., words where there is always an even number of occurrences of 0 between any two occurrences of 1. This shift is sofic, but is not an SFT. The reader may consult the text [52, Chap. 2, Chap. 3] to gain more familiarity with SFTs and sofic shifts.

1.3.2 Measures

A *cylinder* of $X \subseteq \Sigma^\omega$ is a set of the form $\{ux' \colon x' \in \Sigma^\omega\} \cap X$ where $u \in \Sigma^*$. The cylinder generated by u will be denoted by $[u]_X$; we omit the superscript X when it is clear from the context. The cylinders are clopen (i.e., both closed and open, and in particular are the open balls of X), and a subset of X is clopen if and only if it is a finite union of cylinders. Since there are countably many cylinders, any open set is a countable union of cylinders.

We refer the reader to [34, App. B.5] or a standard text such as [9] for the basic concepts of measure spaces. Given a topological space X, its collection of *Borel-measurable sets*, or simply *Borel sets*, is the the σ-algebra generated by the open subsets of X. If X is a shift space, the σ-algebra can equivalently be generated by the cylinders of X.

A (Borel probability) *measure* μ on X maps each Borel set B to $\mu(B) \in [0, 1]$, satisfies $\mu(X) = 1$, and is countably additive, i.e., for any countable collection $(B_n)_{n=0}^{\infty}$ of disjoint Borel sets we have $\mu(\bigcup_n B_n) = \sum_n \mu(B_n)$.

A function $\pi \colon Y \to X$ is called *Borel measurable* if for every Borel set $B \subseteq X$, the pre-image $\pi^{-1}(B)$ is a Borel set of Y. In particular, if π is continuous, then it is Borel measurable. Indeed, if B is open, then by continuity $\pi^{-1}(B)$ is open; and we have that $\pi^{-1}(X \setminus B) = Y \setminus \pi^{-1}(B)$ and $\pi^{-1}(\bigcup_i B_i) = \bigcup_i \pi^{-1}(B_i)$. In other words, the family of sets B for which $\pi^{-1}(B)$ is Borel contains the open sets of X and forms a σ-algebra; in particular it must contain the σ-algebra generated by the family of open sets of X, i.e., the Borel sets of X. A measurable function $\pi \colon Y \to X$ *projects* a Borel measure ν on Y to a Borel measure μ on X as $\mu = \nu \circ \pi^{-1}$.

An *invariant measure* μ of a compact dynamical system (X, T) is a (Borel probability) measure μ such that $\mu(T^{-1}B) = \mu(B)$ for every measurable $B \subseteq X$. By the Krylov-Bogolyubov theorem [34, Thm. 3.8.4], every (X, T) has an invariant measure.

In the case of shift spaces, we can obtain an invariant measure μ by defining a probability pre-measure μ_0 on the Boolean algebra of cylinders, and then invoking the Carathéodory extension theorem [34, Thm. B.5.1] to uniquely extend it to a Borel measure μ, since the σ-algebras generated by cylinders and open sets of a shift space coincide. We require that the pre-measure μ_0 evaluate to 1 on $X = [\varepsilon]_X$, and satisfy the compatibility conditions

$$\mu_0([u]_X) = \sum_a \mu_0([ua]_X) = \sum_a \mu_0([au]_X). \tag{1.12}$$

Observe in particular that $\mu_0 \circ T^{-1}([u]_X) = \mu_0([u]_X)$, and more generally, for any set U in the Boolean algebra of cylinders, $\mu_0(U) = \mu_0(T^{-1}U)$. The Borel measure μ is defined as

$$\mu(B) = \inf_{(U_n)_n} \sum_n \mu_0(U_n),$$

where $(U_n)_n$ ranges over sequences of (disjoint) sets in the Boolean algebra of cylinders such that $B \subseteq \bigcup_n U_n$.

Since T is continuous, $\mu \circ T^{-1}$ will also be a Borel measure ν. The corresponding pre-measure ν_0 obtained by restricting ν to the Boolean algebra of cylinders, however, is identical to μ_0. Thus by the uniqueness of the extension, $\nu = \mu \circ T^{-1} = \mu$, i.e., μ is indeed an invariant measure.

We shall henceforth describe invariant measures by how they evaluate cylinders. For example, in a Sturmian shift X_η, we can define an invariant measure that maps the cylinder $[u]$ to the length of the interval I_u. In the full shift Σ^ω, we can define an invariant measure that evaluates $[u]$ to $\frac{1}{|\Sigma|^{|u|}}$: we call this the *uniform invariant measure*.

We say that $x \in X$ is a *generic point* of an invariant measure μ (or simply, x is generic for μ) if for every continuous $h \in C(X, \mathbb{R})$ (where $C(X, \mathbb{R})$ denotes the set of all continuous functions from X to $\mathbb{R}$),

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} h(T^i x) = \int_X h d\mu. \tag{1.13}$$

We refer to expressions of the form $\frac{1}{N}\sum_{i=0}^{N-1} h(T^i x)$ as *Birkhoff averages*.

We refer the reader to [34, App. B.5.2] for the precise formal definition of the integral; when it is clear from the context, we shall omit the subscript X denoting the domain of the integral and simply write $\int h d\mu$. As an example, normal words are generic points for the uniform invariant measure on the full shift.

Chapter 2

Automata on Recurrent Words

Semënov showed that if a word $x \in \Sigma^\omega$ is effectively almost-periodic then for any transducer $\mathcal{A}$, the word $\mathcal{A}(x)$, if infinite, is also effectively almost-periodic (see [58, Sec. 3] for an exposition of that result in English). In particular, if x is effectively almost-periodic, then Acc_x is decidable. Using this result and techniques from the proof, Pritykin showed that if a word $y \in \Sigma^\omega$ has a uniformly recurrent suffix x, then for any transducer $\mathcal{A}$, the word $\mathcal{A}(y)$, if infinite, also has a uniformly recurrent suffix [65]. Around the same time, Delvenne et al. independently developed a purely symbolic-dynamic proof [26, Prop. 9]. We dedicate this chapter to extensions of these results.

2.1 Preservation of Recurrence

We use the lightweight version of the Krohn–Rhodes theorem (Prop. 1.2.8) to prove (in an arguably simpler and more insightful manner) the following strengthening of the results of Semënov and Pritykin. We remark that the Krohn–Rhodes theory is crucially needed in the case where the recurrence is not necessarily uniform. The case of recurrent words is novel; see also [47, Chap. 3] for a treatment of the almost-periodic case.

Theorem 2.1.1. *Let x be an infinite word, and let $\mathcal{A}$ be a transducer such that $\mathcal{A}(x)$ is also an infinite word. If x possesses one of the following properties, then $\mathcal{A}(x)$ also has that property.*

1. *has a recurrent suffix*
2. *has a uniformly recurrent suffix*
3. *has a linearly recurrent suffix*

4. *is (effectively) almost-periodic*

Furthermore, if the suffix of x is effectively (uniformly, linearly) recurrent and its starting index in x can be computed, then the corresponding suffix of $\mathcal{A}(x)$ also enjoys these properties.

Since we have already shown that the preservation result holds for substitutions (Lems. 1.1.18 and 1.2.4), it suffices to prove the above for automata rather than transducers. Moreover, the action of an automaton can also be decomposed into a letter-to-letter substitution applied to the actions of permutation-reset automata in series (Prop. 1.2.8). Thus, we only need to prove the above statement for permutation-reset automata. We divide the proof into two lemmas: the simpler one assumes the presence of reset letters in x, the other assumes all transitions are permutations and distills the essence of Semënov's argument.

We shall also assume, without losing generality, that the input x is recurrent in the first three cases. One simply adjusts the initial state of the automaton to account for the discarded prefix.

Lemma 2.1.2. *Let $\mathcal{A}$ be a permutation-reset automaton, let x be a word, and let N be an index such that $x(0, N+1)$ contains a reset letter, and let $y = \mathcal{A}(x)$. If x has one of the following properties, then $z = y(N+1, \infty)$ also has that property.*

1. *recurrent*
2. *uniformly recurrent*
3. *linearly recurrent*
4. *almost-periodic*

Furthermore, if these properties are effective for x, then the index N can be computed, and the properties are effective for $y(N+1, \infty)$ as well.

Proof. We begin by showing how N can be computed if the recurrence is effective. In the case of uniform recurrence, one simply takes $N = R_x(1)$. Otherwise, we have in the case of effective recurrence that Acc_x is decidable, and we can determine which reset letters occur in x. We then enumerate the letters of x till we find a reset letter. In the case of effective almost periodicity, the factor language is recursive, and we enumerate the letters of x till we find a reset letter. We shall assume for technical convenience that $x(N)$ is a reset letter: it can readily be checked that this does not lose any generality.

We also note that in the case that x is effectively recurrent, showing that z is recurrent suffices to show that it is, in fact, effectively recurrent. Indeed, we have that Acc_x is decidable, and we can effectively construct a transducer $\mathcal{B}$ such that $z = \mathcal{B}(x)$. By Lem. 1.2.2, we have that Acc_z is decidable.

In the cases of uniform and linear recurrence, we express a return-time function R_z in terms of R_x. In the case of almost periodicity, we additionally give an implementation of V_z.

We start with item (1). We factorise $x(N, \infty)$ as $r_0 r_1 \cdots$, where each $r_i = a v_i$ is a return word to the reset letter a which occurs at index N. We use $n_0, n_1, \ldots$ to denote the starting indices of these factors: these are also precisely the indices at which the letter a occurs in x.

Consider an occurrence of a factor v of $y(N+1, \infty)$ which is obtained as the run on u starting in state q. We shall denote v by $\langle q, u \rangle$. Let the corresponding occurrence of u in x be at index m. To prove that $y(N+1, \infty)$ is recurrent, it would suffice to demonstrate that there is another occurrence of v to the right of the one under consideration.

To do so, we consider the largest i such that $n_i < m$, which is guaranteed to exist since $N = n_0 < m$. Let $x(n_i, m + |u|) = au'u$. When x is recurrent, there exists $j > i$ such that $x(n_j, n_j - n_i + m + |u|) = au'u$. Since a is a reset letter, the automaton $\mathcal{A}$ begins reading the occurrence of u at index $n_j - n_i + m$ in state q: this corresponds to an occurrence of $v = \langle q, u \rangle$ to the right of the one we began with, as desired. This proves item (1).

To prove items (2) and (3), we will give an appropriate expression for a return-time function $R_z(\ell)$, and observe that it can be computed if R_x is computable. We observe from the preceding argument that we can take $R_z(\ell) = R_x(\ell + R_x(1))$, the latter being a bound on $n_j - n_i$. Clearly, if R_x grows at most linearly, then so does R_z.

Finally, we argue that when x is almost-periodic, we can still choose $R_z(\ell) = R_x(\ell + R_x(1))$ as proven above, and we choose $V_z(\ell) = -N + V_x(\ell + R_x(1)) + R_x(1)$. The first requirement on V_z, i.e., finding an occurrence to the right, has already been proven above. The proof of the second requirement is obtained *mutatis mutandis*.

Formally, if v occurs at index $m \geq V_z(\ell) + R_z(\ell)$, the projection u of v occurs in x at index

$$N + m \geq N + V_z(\ell) + R_z(\ell) = V_x(\ell + R_x(1)) + R_x(1) + R_x(\ell + R_x(1)).$$

The factor $au'u$ then occurs in x at index $N+m-k \geq V_x(\ell+R_x(1))+R_x(\ell+R_x(1))$, and hence by the definition of V_x, the factor $au'u$ has another occurrence in x at index $N+m-k-k' \geq N+m-k-R_x(\ell+R_x(1))$, which leads to an occurrence of v in z at index $m-k' \geq m-R_x(\ell+R_x(1)) = m-R_z(\ell)$. This proves item (4). □

Lemma 2.1.3. *Let $\mathcal{A}$ be a permutation automaton. If word x has one of the following properties, then the run $y = \mathcal{A}(x)$ also has that property.*

1. *recurrent*
2. *uniformly recurrent*
3. *linearly recurrent*
4. *almost-periodic*

Furthermore, if these properties are effective for x, they are also effective for $\mathcal{A}(x)$.

Proof. As we did in proving the previous lemma, we observe that if x is effectively recurrent, then Acc_x is decidable, and hence Acc_y would be decidable. Thus, proving y to be recurrent would also prove it to be effectively recurrent in this case.

In the cases of uniform and linear recurrence, we express a return-time function R_y in terms of R_x, and observe that it can be computed if R_x is computable. In this case, it will be more convenient to also work with window functions; recall that $W_x(\ell) = R_x(\ell)+\ell$. In the case of almost periodicity, we additionally give an implementation of V_y.

We start with item (1). As before, given an occurrence of $v = \langle q, u_0\rangle$ (i.e., v is the run on u_0 starting in state q) in y at index n_0, we seek another occurrence to its right. We shall iteratively use the recurrence of x for this purpose. We shall inductively define factors $u_0, u_1, \ldots$ of x, each occurring at index n_0. The word u_{i+1} is of the form $r_i u_i$, where r_i is the return word to u_i that occurs at index n_0. We also define $q_i = \delta(q, u_i)$.

By the pigeonhole principle, there exist $0 \leq i < j \leq |Q|$ such that $q_i = q_j$. By construction, u_0 is both a prefix and a suffix of u_i, which in turn is both a proper prefix and a proper suffix of u_j. Observe the occurrences of u_i at distinct indices n_0 and $n_0+|u_j|-|u_i|$. Both these occurrences lead to state q_i. Since $\mathcal{A}$ is a permutation automaton, this is only possible if the runs on both occurrences began in the same state, i.e., in q. This means that we have found an occurrence of $v = \langle q, u_1\rangle$ at index $n_0+|u_j|-|u_i|$. This proves item (1).

To prove items (2) and (3), let us bound the length $|u_j| - |u_i|$ of the return word in the case that x is uniformly recurrent. It suffices to bound $|u_{|Q|}|$. Recall the window function, and observe that $|u_{i+1}| \leq W_x(|u_i|)$. A straightforward induction shows that $|u_k| \leq W_x^k(|u_0|)$, where W_x^k denotes the k-fold composition of W_x. Thus, we let $W_y(\ell) = W_x^{|Q|}(\ell)$: this bounding function is linear if R_x is, and also computable if R_x is. This establishes items (2) and (3).

Finally, when x is almost-periodic, we claim that we can still choose $W_y(\ell) = W_x^{|Q|}(\ell)$ as above, and that we can choose $V_y(\ell) = V_x \circ W_x^{|Q|-1}(\ell)$. Indeed, let v occur in y at index $m \geq V_y(\ell)$. We thus have that $u_0, u_1, \ldots, u_{|Q|-1}$ are recurrent, and moreover for each of these words u_i, the factor u_i of length $W_x^i(\ell)$ occurs at index m, and subsequently at index at most $m + R_x(u_i)$. The reasoning in the preceding paragraphs to find an occurrence of v to the right thus applies as is.

To complete the proof, we observe that the same reasoning, albeit with slightly more bookkeeping, works to find an occurrence of v to the left when $m \geq V_y(\ell) + R_y(\ell)$. It is helpful to recall from (1.10) that

$$R_y(\ell) = W_y(\ell) - \ell = W_x^{|Q|}(\ell) - \ell = \sum_{i=0}^{|Q|-1} R_x \circ W_x^i(\ell),$$

and hence

$$m \geq V_x \circ W^{|Q|-1}(\ell) + \sum_{i=0}^{|Q|-1} R_x \circ W_x^i(\ell).$$

Let the length-ℓ factor $v = \langle q_0, u_0 \rangle$ occur at index $m = m_0$. We shall inductively define factors $u_0, u_1, \ldots$ of x and indices $m_1, m_2, \ldots$, such that m_{i+1} is the index of the occurrence of u_i immediately before the one at m_i, and $u_{i+1} = x(m_{i+1}, m_i + |u_i|) = x(m_{i+1}, m_0 + |u_0|)$. We also define q_i to be the state the automaton is in at index m_i.

We check, by a simple induction, that for $i \in \{0, 1, \ldots, |Q| - 1\}$ we have that $|u_i| \leq W_x^i(\ell)$, the index m_i is at least $V_x \circ W_x^{|Q|-1}(\ell) + \sum_{j=i}^{|Q|-1} R_x \circ W_x^i(\ell)$, and hence by the defining property of V_x, the invariant holds for u_{i+1} and m_{i+1}. In particular, $|u_{|Q|}| \leq W_x^{|Q|}(\ell)$, and $m_{|Q|} \geq V_y(\ell)$.

We find a leftward occurrence of $\langle q_0, u_0 \rangle$ by the same pigeonhole argument as before. Let $i < j$ be such that $q_i = q_j$. We use the fact that u_0 is a suffix of u_i, which in turn is a proper prefix of u_j. Thus, our desired occurrence of v is at index $m_j + |u_i| - |u_0| = m_0 - (|u_j| - |u_i|)$. The gap is bounded by $|u_{|Q|}| - \ell \leq W_x^{|Q|} - \ell = R_y(\ell)$, as desired. This completes the proof of item (4). □

We record corollaries that are helpful for verification: the first two follow immediately from Thm. 2.1.1, and the last is a corollary of the proof.

Corollary 2.1.4. *If x is an effectively almost-periodic word, then Acc_x is decidable.*

Proof. Given a deterministic parity automaton $\mathcal{A}$, let $y = \mathcal{A}(x)$. We record the states occurring in the $R_y(1)$ consecutive indices starting from $V_y(1)$, and check whether the highest colour amongst these states is even. □

Corollary 2.1.5. *If X is a minimal shift, we can decide, given an ω-regular language L, whether $X \cap L = \{\}$.*

Proof. We simply run the algorithm we have developed on all possible factors of length $R_y(1) + V_y(1)$ that can occur in X. Here, R_y, V_y are functions that are expressed in terms of R_x, which is the same for all $x \in X$. □

The following corollary is useful while studying the acceptance problem for classes of words with a fixed factor complexity, e.g., Sturmian words, or Arnoux-Rauzy words.

Corollary 2.1.6. *Let $x \in \Sigma^\omega$ be effectively uniformly recurrent and let p_x be its computable factor-complexity function. Given any deterministic parity automaton $\mathcal{A}$ (equivalently, any ω-regular language L), we can compute N with the following property. For any uniformly recurrent word $y \in \Sigma^\omega$ such that $y(0, N) = x(0, N)$ and $p_y = p_x$, we have that $\mathcal{A}$ accepts x if and only if it accepts y (equivalently, $x \in L$ if and only if $y \in L$).*

Proof. We will show that if x and y have the same factor-complexity function, and agree on a long enough prefix, then the information we use to decide whether $\mathcal{A}$ accepts a uniformly recurrent word is identical for x and y.

Let $z = \mathcal{A}(x)$, and $w = \mathcal{A}(y)$. We have shown that in order to determine acceptance, we need to read $z(V_z(1), V_z(1) + R_z(1))$. When x is uniformly recurrent, we can moreover take $V_z(\ell) = M$ for all ℓ, where M is such that $z(M, \infty)$ is uniformly recurrent. In the proof of Thm. 2.1.1, we have shown that V_z and R_z can be computed via a fixed expression that is purely in terms of R_x, e.g., when $\mathcal{A}$ is decomposed into a permutation automaton with k states followed by a reset automaton, $R_z(\ell) = W_x^k(\ell + W_x^k(1) - 1) - (\ell + W_x^k(1) - 1)$. In other words, the expressions for V_w, R_w are obtained from those for V_z, R_z by simply replacing R_x with R_y.

Let ℓ_x be the largest argument supplied to R_x while computing $V_z(1)$ and $R_z(1)$. We will show how R_x can be computed by enumerating prefixes of x. This will prove that if x and y have a long enough prefix in common, say of length N', then R_x and R_y will agree for all $\ell \leq \ell_x$, and hence $V_z(1) = V_w(1)$, and $R_z(1) = R_w(1)$. It would then remain to take N to be the maximum of N' and $V_z(1) + R_z(1)$.

We enumerate $\mathcal{L}_\ell(x)$ by computing a prefix of x long enough to have $p_x(\ell)$ distinct factors. For any factor u, we compute the set of its return words by enumerating $\mathcal{L}(x)$ till we find a length ℓ' such that every factor of this length contains at least two occurrences of u. We take $W_x(\ell)$ to be the maximum such length ℓ' over all factors u of length ℓ. Thus, given ℓ, let $N(\ell)$ denote the length of the prefix of x computed in order to determine $W_x(1), \ldots, W_x(\ell)$. The length N' required in the preceding paragraph is then simply $N(\ell_x)$. This completes the proof: indeed, the information about x we accessed to determine whether $\mathcal{A}$ accepts it is identical for y. □

2.2 Prefix-Independent Verification

In this section, we discuss a variant of Thm. 2.1.1 for (not necessarily effective) almost-periodic words and *prefix-independent* ω-regular languages. Such languages have been considered for the verification of dynamical systems in [4, 6]. Formally, a language $L \subseteq \Sigma^\omega$ is prefix-independent if for all $u, v \in \Sigma^*$, and $x \in \Sigma^\omega$, we have that $ux \in L$ if and only if $vx \in L$. Clearly, if a nonempty language L is prefix-independent, then it is dense in Σ^ω. (In verification parlance, we refer to languages that are dense in Σ^ω as *liveness properties*.) However, the dense language $(\Sigma\Sigma)^*(a\Sigma)^\omega$ is not a prefix-independent language.

As further examples, the language $\Sigma^*(a\Sigma)^\omega$ is prefix-independent but not definable in first-order logic; the language $b^\omega + (b^*a)^\omega$ is dense, first-order definable, but not prefix-independent: we have that b^ω is in the language but ab^ω is not.

We record two structural properties of prefix-independent languages that might be of independent interest. The first of these follows directly from Lem. 1.2.5.

Proposition 2.2.1. *Let $L \subseteq \Sigma^\omega$ be a prefix-independent ω-regular language. Either all disjunctive words over Σ are contained in L, or all disjunctive words are excluded from L. Furthermore, we can determine which is the case.*

Indeed, prefix-independence necessitates that all bottom strongly connected components of the recognising automaton $\mathcal{A}$ agree on the parity of the highest colour within them.

Proposition 2.2.2. *An ω-regular language $L \subseteq \Sigma^\omega$ is prefix-independent if and only if $L = \bigcup_{i=1}^{d} \Sigma^* V_i^\omega$, where $V_1, \ldots, V_d$ are regular languages that do not contain the empty word.*

Proof. We assume without losing generality that L is non-empty.

Suppose $L = \bigcup_{i=1}^{d} \Sigma^* V_i^\omega$. Suppose, for the sake of contradiction, there exists a pair $x = wy, x' = w'y$ of words such that $x \in \Sigma^* V_i^\omega \subseteq L$ but $x' \notin L$. We have that $x = uv_1v_2\ldots$, where for each $n \geq 1$, $v_n \in V_i$. Since x' has the same suffix y as x, it can be factorised as $u'v_mv_{m+1}\ldots$, which witnesses that $x' \in \Sigma^* V_i^\omega$, a contradiction. Thus $L = \bigcup_{i=1}^{d} \Sigma^* V_i^\omega$ implies that L is prefix-independent.

Conversely, suppose $L = \bigcup_{i=1}^{d} U_i V_i^\omega$ is prefix-independent. Clearly, $L \subseteq \bigcup_{i=1}^{d} \Sigma^* V_i^\omega$. We shall prove that the reverse inclusion also holds. To that end, consider $x = wv_1v_2\cdots \in \Sigma^* V_i^\omega \subseteq \bigcup_{i=1}^{d} \Sigma^* V_i^\omega$, and $x' = uv_1v_2\cdots \in U_i V_i^\omega \subseteq L$. Since L is prefix-independent and $x' \in L$, we must have that $x \in L$, and hence $L = \bigcup_{i=1}^{d} \Sigma^* V_i^\omega$. □

The main result of this section is a slightly generalised reformulation of [4, Sec. 5], with simplified techniques. (We leave gauging the extent of simplification to the discernment of the reader.)

Proposition 2.2.3. *Let $X \subseteq \Sigma^\omega$ be the set of almost-periodic words x such that we have $\mathcal{L}^{\mathsf{rec}}(x) = \mathcal{L}_\star$. Let $L \subseteq \Sigma^\omega$ be a prefix-independent ω-regular language. We have that either $X \subseteq L$, or $X \cap L = \{\}$. Furthermore, if $\mathcal{L}_\star$ is recursive, we can decide which of the two cases holds.*

Note that if we replace "almost-periodic" by "uniformly recurrent" then X will be a minimal shift. It is not the case, however, that all words of a minimal shift must agree upon membership in any arbitrary prefix-independent language: e.g., the set of words that admit factor frequencies is a prefix-independent language, and there exist minimal shifts where "most" but not all words admit factor frequencies.

Proof of Prop. 2.2.3. We will show that for any $x \in X$, we can decide whether $x \in L$ purely with oracle queries to $\mathcal{L}_\star$. Throughout this proof, we assume that L is recognised by a deterministic parity automaton $\mathcal{A} = (Q, \Sigma, q_{\mathsf{init}}, \delta, \mathsf{col})$, and all states in $\mathcal{A}$ are reachable from q_{init}. For technical convenience, we define a function $C : Q \times \Sigma^* \to \mathbb{N}$ to map (q, u) to the maximum colour seen along the run on u starting in state q. In other words, if the run fragment is $(q_0, a_0)\cdots(q_{|u|-1}, a_{|u|-1})$, then $C(q_0, u) = \max\{\mathsf{col}(q_0), \ldots, \mathsf{col}(q_{|u|-1})\}$.

Recall from Lem. 1.1.10 that for any factor $u \in \mathcal{L}_\star$, we can express any $x \in W$ as the concatenation $vr_1r_2\cdots$, where $r_1, r_2, \ldots$ are recurrent return words to u. We can indeed enumerate $\mathcal{R}_x^{\mathsf{rec}}(u)$, the set of recurrent return words to u by enumerating words in $\mathcal{L}_\star$ till we find a length ℓ such that all factors of this length have at least two occurrences of u.

We shall iteratively find a *saturated* factor $u \in \mathcal{L}_\star$ such that for all recurrent return words r to u and all states $q \in Q$, we have that $C(q,u) = C(q,ru)$. Let $q_u = \arg\min_q C(q,u)$, and observe that if $\mathcal{A}$ starts reading the suffix $r_1 r_2 \cdots$ in state q_u, then the highest colour seen infinitely often is $C(q_u, u)$.

Indeed, the highest colour seen while reading each of the infinitely many occurrences of u is at least $\min_q C(q,u)$; however by saturation, the highest colour seen while reading any word ru that precisely spans two consecutive occurrences of u must have already been seen while reading the first occurrence of u. These constraints force the highest recurrent colour to be $C(q_u, u) = \min_q C(q,u)$.

Since q_u is reachable from q_{init}, there is a word w such that $\delta(q_{\mathsf{init}}, w) = q_u$. By the prefix independence of L, we have that $x \in L$ if and only if $w r_1 r_2 \cdots \in L$, if and only if $\min_q C(q,u)$ is even.

It remains to compute a saturated factor u. We define a quasi-order on Σ^+ for our fix-point procedure as follows. We declare $u \preceq v$ if for all q we have that $C(q,u) \leq C(q,v)$. Observe that if u is a prefix of v, then $u \preceq v$, and that this quasi-order has at most $|Q|^2$ equivalence classes. To find a saturated factor, we simply start with an arbitrary factor u_0 and iterate till saturation. If u_i is not saturated, there exists one r_i among its finitely many recurrent return words, and a state q such that $C(q,u_i) < C(q, r_i u_i)$. We set $u_{i+1} = r_i u_i$. This process will terminate in at most $|Q|^2$ iterations. □

The same proof based on saturated factors works to establish the following.

Corollary 2.2.4. *Let $X \subseteq \Sigma^\omega$ be the set of all recurrent words x such that we have $\mathcal{L}(x) = \mathcal{L}_\star$. Let $L \subseteq \Sigma^\omega$ be a prefix-independent ω regular language. We have that either $X \subseteq L$, or $X \cap L = \{\}$. We can determine which is the case if we can furthermore decide, given any regular language $L' \subseteq \Sigma^+$, whether $\mathcal{L}_\star \cap L' = \{\}$.*

The additional effectiveness assumption circumvents the obvious hurdle of the infinitude of (recurrent) return words that arises in implementing the iterative search for a saturated factor. Indeed, when u is fixed, the set of words ru that (i) contain precisely two occurrences of u, once as a prefix, and once as a suffix, and (ii) satisfy $C(q,ru) > C(q,u)$ for some state q, is a regular language. The factor u is saturated if and only if the intersection of this language with $\mathcal{L}_\star$ is empty. (If the intersection is non-empty, an element for the next iteration can be found by brute enumeration.)

It is tempting to state that the above corollary generalises to arbitrary words x if we replace $\mathcal{L}(x)$ with $\mathcal{L}^{\mathsf{rec}}(x)$. This, however, is not the case. The prefix-independent language $\{0,1\}^*(0(00)^*1)^\omega$, which is the set of words such that the letter 1 eventually

occurs exclusively at positions of the same parity, distinguishes between the words $0^2 10^4 10^6 1 \cdots$ and $010^3 10^5 1 \cdots$, which have the same recurrent factor language $0^* + 0^* 10^*$.

We can prove that the above corollary cannot be made effective if we only assume that $\mathcal{L}_\star$ is recursive.

Proposition 2.2.5. *There exists a recurrent word x such that (i) x is effective, (ii) $\mathcal{L}(x)$ is recursive, but (iii) it is undecidable, given a prefix-independent ω-regular language L, whether $x \in L$.*

Proof. We construct x such that its factors encode information about the runs of Turing machines. To that end, let the alphabet $\Sigma = \{a, b, c, h\}$, and fix a Gödel numbering on Turing machines (with the starting index being 1). We encode the information "the m-th Turing machine halts (on empty input) in k steps" as $a^m b^k h$, and we encode "the m-th Turing machine does not halt after having run for k steps" as $a^m b^k c$. The letters c, h are thus delimiters respectively standing for "continues" and "halts".

We let $L_{\mathsf{TM}} \subseteq \Sigma^+$ be the set of valid encodings. If the m-th machine indeed halts in k steps, then it contributes the words $a^m c, a^m bc, \ldots, a^m b^{k-1} c, a^m b^k h$ to L_{TM}; if the m-th machine does not halt, then it contributes $a^m b^* c$.

We shall construct x such that $\mathcal{L}(x)$ is the set of factors of words in L_{TM}^+. Let us show that this language is recursive. If a word has no delimiters, then it is in the language if and only if it is of the form $a^* b^*$. Given a word $u \in \Sigma^+$ such that the first delimiter is at position i and the last at position j, we first check whether $u(i+1, j+1)$ is a word in L_{TM}^*. Next, we check that the suffix $u(j+1, |u|)$ is admissible: it should either be empty, or be of the form $a^m b^k$, where $m \geq 1$ and the m-th Turing machine runs for at least k steps. Finally, we observe that the prefix $u(0, i+1)$, as long as it is of the form $a^* b^* \{c, h\}$, will always be admissible because there are infinitely many Turing machines that do not halt, and for any k, there are infinitely many Turing machines that halt in precisely k steps.

We are now ready to construct $x \in L_{\mathsf{TM}}^\omega$ using Lem. 1.1.7. We fix an enumeration of L_{TM}^+, and assign these words to $v_0, v_1, \ldots$. We take an arbitrary word in L_{TM} as u_0, define $u_{n+1} = u_n v_n u_n$ for all n, and construct $x = \lim_{n \to \infty} u_n$.

Finally, observe that the m-th Turing machine halts if and only if

$$x \in (\Sigma^* \{c, h\} a^m b^* h)^\omega.$$

This proves the desired undecidability. □

2.3 Effective Recurrence Revisited

In this section, we apply the "anything that can happen, will happen" principle of Lem. 1.2.5 to prove that the result of Cor. 2.2.4 can be strengthened to arbitrary ω-regular languages for recurrent words, and to incidentally prove Semënov's Thm. 1.2.3.

As the first step, we will prove the result assuming that the recurrent word x is typical for a sofic shift, i.e., has regular factor language. We subsequently show that for any recurrent word x and ω-regular language L, there is a "representative word" y with regular factor language and an ω-regular language L' such that $x \in L$ if and only if $y \in L'$. Our effectiveness assumptions on x allow us to compute y and present its regular factor language.

We remark that our proof of the former formulates the ideas of [52, Chap. 3] in algebraic terms,[1] and briefly recall the relevant notions. Let $\mathcal{L}(x) \neq \Sigma^*$ be the regular factor language of some word $x \in \Sigma^\omega$. We have that it is recognised by a morphism into a finite syntactic monoid M, i.e., we have a morphism $h : \Sigma^* \to M$ and a subset $F \subseteq M$ be such that $u \in \mathcal{L}(x)$ if and only if $h(u) \in F$. The minimality of the syntactic monoid implies that $h^{-1}(m) \neq \{\}$ for every $m \in M$, and that for every pair m_1, m_2 of distinct elements of F, there exists $m \in F$ such that $m_1 m \in F$ but $m_2 m \notin F$. (A dual statement holds for left multiplication too.)

The factor language is (i) factorial and (ii) extendable, i.e., (i) if u is a factor of u' and $u' \in \mathcal{L}(x)$, then $u \in \mathcal{L}(x)$, and (ii) for every $u \in \mathcal{L}(x)$, there exist letters a, b such that $\{au, ub\} \subseteq \mathcal{L}(x)$. The first property assures us that $M \setminus F = \{0\}$, where the absorbing element 0 is such that $0 \cdot m = m \cdot 0 = 0$ for all m. Thus, if $m_1 \cdot m_2 \in F$, then $\{m_1, m_2\} \subseteq F$. The second property implies that for every $m \in F$, there exist elements $m_1, m_2 \in F$ such that $\{m_1 m, m m_2\} \subseteq F$.

Proposition 2.3.1. *Let the recurrent word $x \in \Sigma^\omega$ be effective, and let $\mathcal{L}(x)$ be recursive by virtue of being regular. We have that Acc_x is decidable.*

Proof. We shall compute a prefix u_0 of x, construct a transducer $\mathcal{A}$, and define an effective disjunctive word y such that $x = u_0 \cdot \mathcal{A}(y)$. The claim would then follow immediately from Lem. 1.2.2 (Acc_x reduces to Acc_y) and Lem. 1.2.5 (Acc_y is decidable).

Let M be the computable finite syntactic monoid of $\mathcal{L}(x)$, and let the morphism $h : \Sigma^* \to M$ and $F \subseteq M$ be such that $u \in \mathcal{L}(x)$ if and only if $h(u) \in F$.

[1] Lind and Marcus acknowledge that their exposition borrows from automata theory. Weiss, who introduced sofic shifts as the closure of SFTs under (letter-to-letter) transduction, himself described them in terms of semigroups. The term "sofic" is derived from the Hebrew word for "finite".

We claim that there exists a *synchronising word* $w_0 \in \mathcal{L}(x)$ such that $F \cdot h(w_0) = \{m_0\}$. Indeed, for any word $u \in \mathcal{L}(x)$, let $\mathsf{targets}(u)$ denote the set $F \cdot h(u) \cap F$. If this set contains two distinct elements m_1, m_2, we use the minimality of the syntactic monoid to find a word v such that $m_1 \cdot h(v) \in F$, but $m_2 \cdot h(v) \notin F$. The extendability of the factor language guarantees that $uv \in \mathcal{L}(x)$ because we have that $m_1 \cdot h(v) = m_3 \cdot h(u) \cdot h(v)$ for some $m_3 \in F$; equivalently we have that $m_3 \cdot h(uv) \in F$, which necessitates $h(uv) \in F$. Crucially, we have that $|\mathsf{targets}(uv)| < |\mathsf{targets}(u)|$. We iterate in this manner till we find a word w_0 such that $|\mathsf{targets}(w_0)| = 1$, as desired.

We define our required prefix u_0 of x to the shortest prefix that contains an occurrence of w_0. Our disjunctive word will be over the alphabet $\Gamma = \{0, \ldots, |\Sigma| - 1\}$, and for each $m \in F$, we define a surjective function

$$\mathsf{letter}_m : \{0, \ldots, |\Sigma| - 1\} \to \{a \in \Sigma \mid m \cdot h(a) \in F\}$$

as follows. Let the cardinality of the image be c, and fix an order on its elements. We map i to the $(i \bmod c)$-th element.

Let us now construct our transducer $\mathcal{A}$ as $(F, m_0, \Gamma, \delta, \tau)$, i.e., the elements of F serve as the states, the synchronising element m_0 is the initial state, we have $\delta(m, i) = m \cdot h(\mathsf{letter}_m(i))$, and $\tau(m, i) = \mathsf{letter}_m(i)$. Any word y that satisfies $y(n) \in \mathsf{letter}_{m_n}^{-1}(x(|w_0| + n))$ for all n, where $m_n = m_0 \cdot h(x(0, n))$ also satisfies $\mathcal{A}(y) = x(|w_0|, \infty)$ by construction. It is thus straightforward to make a "default" choice of $y(n)$ in the absence of any constraints.

In order to construct y to be effective and disjunctive, let us observe that the image of the function which maps the word $v \in \Gamma^*$ to the word $\mathcal{A}(v)$ is the infinite set $\mathsf{follower}(w_0) = \{u \in \Sigma^* \mid m_0 \cdot h(u) \in F\}$.

We fix an enumeration of $\Gamma^+ \times \mathbb{N}$, and iterate over it to construct increasingly long prefixes of y. We shall maintain the invariant that at the end of each iteration, if the pair under consideration was (v, m), then the constructed prefix $y(0, N)$ must contain at least m occurrences of v. If, at the start of the iteration, we do not have the requisite number of occurrences of v, we consider the factor $w_0 \cdot u$ of x, where $u = \mathcal{A}(v) \in \mathsf{follower}(w_0)$. This factor is recurrent, and by the synchronising property of w_0, we are guaranteed that the suffix u will be read by $\mathcal{A}$ starting in state m_0. Our strategy is simple. We make the default choice for letters $y(n)$, unless the corresponding indices in $x(|w_0|, \infty)$ contain an occurrence of u that is produced starting in state m_0: in this case, we choose letters in a way that produces the word v. In the limit, the effective word y will thus have infinitely many occurrences of every $v \in \Gamma^\omega$, i.e., be disjunctive, and will satisfy $\mathcal{A}(y) = x$, as desired. □

A technically advanced reader might find the above discussion reminiscent of the Krieger Embedding Theorem [48]. In our case, we showed that a particular typical point of a sofic shift can be expressed as the image of a typical point of the full shift.

Proposition 2.3.2. *Let $x \in \Sigma^\omega$ be a recurrent word such that x is effective, and we can decide, given any regular language L, whether $\mathcal{L}(x) \cap L = \{\}$. We have that Acc_x is decidable.*

Proof. We assume that the input ω-regular language to Acc_x is given as an ω-semigroup (M_f, M_ω), morphism $h : \Sigma \to M_f$, and accepting set $F \subseteq M_\omega$, i.e., $x \in L$ if and only if $h(x) \in F$. We shall compute a recurrent word y and an ω-regular language L' such that $\mathcal{L}(y)$ is regular, and $x \in L$ if and only if $y \in L'$. The theorem would then follow immediately from Prop. 2.3.1.

Our alphabet Γ for y is the set of elements of M_f, and we map Γ^* to the monoid $M'_f = M_f \cup \{1_M\}$ using the natural product morphism π that maps a sequence $m_0 \cdots m_{\ell-1}$ to the product of these elements. (We also extend h to map the empty word to 1_M.) We define the language $L' = \{w \in \Gamma^\omega \mid \pi(w) \in F\}$. We now construct y in a way that ensures $\pi(y) = h(x)$, and that $\mathcal{L}(y)$ is regular.

Using the premise of the theorem, we compute the set

$$\mathsf{images}_x = \{m \in M_f \mid \mathcal{L}(x) \cap h^{-1}(m) \cap \Sigma^+ \neq \{\}\}.$$

Note that if the intersection $\mathcal{L}(x) \cap h^{-1}(m) \cap \Sigma^+$ is nonempty, we can also find an element in it by brute enumeration of words in the regular language $h^{-1}(m) \cap \Sigma^+$.

We shall construct y so that

$$\mathcal{L}(y) = L_\star = \{v \in \Gamma^* \mid \exists m \in \mathsf{images}_x, m_{\mathsf{left}}, m_{\mathsf{right}} \in M'_f.\ m = m_{\mathsf{left}} \cdot \pi(v) \cdot m_{\mathsf{right}}\}.$$

For the technically advanced reader familiar with Green's relations, the above definition simply says that $L_\star$ is the set of all words v such that $\pi(v)$ is $\mathcal{J}$*-above* some element of images_x.

We construct y by representing factors u of x using strings v such that $h(u) = \pi(v)$. By default, we represent a letter a by $h(a)$. In order to ensure that all words in $L_\star$ occur as factors infinitely often, we define a computable function $\mathsf{representative} : L_\star \to \mathcal{L}(x)$ that maps ε to ε, and a word v to a factor u of x such that $h(u) = m_{\mathsf{left}} \cdot \pi(v) \cdot m_{\mathsf{right}}$ for some $m_{\mathsf{left}}, m_{\mathsf{right}} \in M'_f$. We then proceed to construct y from x in a manner similar to the proof of Prop. 2.3.1: fix an enumeration of $L_\star \times \mathbb{N}$; if the pair at the current iterate is (v, n), extend the computed prefix, representing each subsequent occurrence of $\mathsf{representative}(v)$ by the word $m_{\mathsf{left}} \cdot v \cdot m_{\mathsf{right}}$, until there are at least n occurrences

of v. Note that if either of $m_{\mathsf{left}}, m_{\mathsf{right}}$ is 1_M, we omit it. Clearly by construction, all elements of $L_\star$ occur infinitely often in y.

We complete the proof by showing that $\mathcal{L}(y) \subseteq L_\star$. Indeed, let v be a factor of y. By construction, there exists a factor u of x, and factors $v_{\mathsf{left}}, v_{\mathsf{right}}$ such that $h(u) = \pi(v') = \pi(v_{\mathsf{left}})\pi(v)\pi(v_{\mathsf{right}})$. By definition, $v \in L_\star$, as desired. □

The reader who is familiar with Colcombet's proof [22] of Semënov's Thm. 1.2.3 will find the preceding propositions wholly unsurprising. While it is technically involved to give a precise translation, we do remark that intuitively, the main role of the subroutine that decides whether $x \in L\Sigma^\omega$ is to identify a suffix x' such that if there is a factor u of x' such that $h(u) = m$, then such factors occur at infinitely many indices. When we are promised that x is itself recurrent, there is no need for this subroutine.

Through this intuition, we incidentally arrive at our own proof of Thm. 1.2.3 by slightly adapting the proof of Prop. 2.3.2. Given a word x, condition (1) that decides whether $x \in L\Sigma^\omega$ ensures that x is effective. Now, given an ω-semigroup (M_f, M_ω), a morphism h, and an accepting set $F \subseteq M_\omega$, when we have to decide whether $h(x) \in F$, we use condition (2) to compute the set of *recurrent elements* $m \in M_f$ such that $x \in (\Sigma^* h^{-1}(m))^\omega$. We then use condition (1) iteratively on increasingly long prefixes until we find a prefix w such that $x \notin w\Sigma^* h^{-1}(m')\Sigma^\omega$ for all nonrecurrent elements m'. We express $x = wx'$, and adapt the construction of Prop. 2.3.2 to x' to construct $y = h(w) \cdot \mathsf{sofic}(x')$ such that the suffix $\mathsf{sofic}(x')$ of y is effectively recurrent with regular factor language whose presentation is computable.

The need to adapt is to account for the possibility that we may not be able to find a recurrent representative for a factor $v \in M_f^*$. Nevertheless, if we are looking for a factor u such that $h(u) = m$, and we have read the prefix w of x so far, we determine whether u can be found starting at the current index by querying whether $x \in wh^{-1}(m)\Sigma^\omega$. In other words, condition (1) compensates for the factors themselves not being recurrent.

Our development allows us to prove the following "prefix-independent" version of Semënov's Thm. 1.2.3. A proof may also be elicited from [22, Cor. 3].

Proposition 2.3.3. *Let $x \in \Sigma^\omega$ be such that we can decide, given any regular language $L \subseteq \Sigma^+$, whether $x \in (\Sigma^* L)^\omega$. We can then decide, given any prefix-independent ω-regular language $L' \subseteq \Sigma^\omega$, whether $x \in L'$.*

Note that we do not require x to be effective.

Proof. As before, we consider the ω-semigroup (M_f, M_ω), morphism h and accepting set F that recognise L', and note that there exists a word $y' = m \cdot y \in M_f^\omega$ such that $\pi(y') = h(x)$, and y is recurrent with regular factor language. While our premise does not guarantee that we can determine m or the letter at a specific index in y, the above discussion can still be applied to compute (finite syntactic monoid of) the factor language of y. Since L' is prefix independent, we even have that $x \in L'$ if and only if $\pi(y) \in F$, i.e., we have reduced our given instance to an instance of checking whether the recurrent word y with a computable regular factor language is in the prefix-independent language $L'' = \{z \in M_f^\omega \mid \pi(z) \in F\}$.

We now apply Cor. 2.2.4. We have that $y \in L''$ if and only if any recurrent word that has the same factor language is in L''; since we can compute the finite syntactic monoid of $\mathcal{L}(y)$, the premise of Cor. 2.2.4 is met. We can hence decide whether $y \in L''$, and equivalently, whether $x \in L'$, as desired. □

Chapter 3

Preservation of Factor Frequencies

In this chapter, we use the ergodic theory of shift spaces to study words that admit factor frequencies. The central consideration of this chapter is as follows: for which words x does the infinite word $\mathcal{A}(x)$ admit factor frequencies for all transducers $\mathcal{A}$?

Note that uniform recurrence will be a common working assumption. Indeed the morphic word $2110^6 1^{18} 0^{54} \cdots$ from the preliminaries is an example of a word that does not admit factor frequencies, and it can be obtained by letter-to-letter transduction from the morphic word $21010^5 10^{17} 10^{53} 1 \cdots$ using a two-state permutation automaton, or from the morphic word $32010^5 20^{17} 10^{53} 2 \cdots$ (which is the fixed point of $3 \mapsto 320, 2 \mapsto 100, 1 \mapsto 200, 0 \mapsto 000$) using a reset automaton. Both these input words admit factor frequencies.

We shall begin by relating the existence of factor frequencies with the ergodic-theoretic notion of genericity. This will motivate us to consider *uniquely ergodic* shifts. We shall then observe that unique ergodicity is not necessarily preserved by the action of a transducer, and thus focus on shifts that satisfy *Boshernitzan's condition*, which implies unique ergodicity. We will finally prove that this stronger condition is preserved by the action of a transducer.

3.1 Factor Frequencies and Genericity

The following lemma states the connection between factor frequencies and ergodic theory.

Lemma 3.1.1. *A word $x \in \Sigma^\omega$ admits factor frequencies if and only if x is generic for an invariant measure μ on the shift X that it generates. Should this be the case, we have $f_x(u) = \mu([u]_X)$ for every factor $u \in \Sigma^*$.*

Proof. The "if" implication is obvious: we simply choose the continuous function h to be the indicator $\mathbf{1}_u$ which evaluates to 1 on the cylinder $[u]_X$ and 0 elsewhere.

Conversely, suppose x admits factor frequencies and generates the shift X. Recall that $\mathcal{L}(X) = \mathcal{L}(x)$, and hence the factor frequencies of x defines a pre-measure μ_0 on each cylinder in a way that satisfies the compatibility condition (1.12). The pre-measure μ_0 is thus invariant, and extends to an invariant measure μ. We now need to prove that x is generic for μ, i.e., for every continuous function h, we have $\int h d\mu = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} h(T^i x)$.

We use an argument reminiscent of the Stone-Weierstraß theorem. Observe that since X is compact, the function h is uniformly continuous, and admits a modulus of continuity. We can define the following sequence $(g_M)_M$ of functions that converges uniformly to h (i.e., for every δ there exists M_δ such that for all $M \geq M_\delta$ we have $\max_X |g_M - h| \leq \max_{u\in\mathcal{L}_M(X)}(\max_{[u]_X} h - \min_{[u]_X} h) < \delta$):

$$g_M = \sum_{u\in\mathcal{L}_M(X)} \max_{[u]_X} h \cdot \mathbf{1}_u.$$

By the dominated convergence theorem [34, Thm. B.5.3], we also have that $\int h d\mu = \lim_{M\to\infty} \int g_M d\mu$. In particular, for every $\delta > 0$, we can choose M large enough such that

$$\left| \int h d\mu - \int g_M d\mu \right| < \delta/3,$$

and for all N, we have by (convexity and) uniform convergence that

$$\left| \frac{1}{N} \sum_{i=0}^{N} g_M(T^i x) - \frac{1}{N} \sum_{i=0}^{N} h(T^i x) \right| < \delta/3.$$

Since x admits factor frequencies, it is also clear that for each M, we have $\lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} g_M(T^i x) = \int g_M d\mu$, and in particular for our chosen M and δ, there exists N_δ such that for all $N \geq N_\delta$, we have

$$\left| \int g_M d\mu - \frac{1}{N} \sum_{i=0}^{N-1} g_M(T^i x) \right| < \delta/3.$$

Adding our inequalities together and applying the triangle inequality, we get that for every δ there exists N_δ such that for all $N \geq N_\delta$, we have

$$\left| \int h d\mu - \frac{1}{N} \sum_{i=0}^{N} h(T^i x) \right| < \delta,$$

or in other words, $\int h d\mu = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} h(T^i x)$, as desired. □

We now describe measures for which "almost" all points are generic. An invariant measure μ is called *ergodic* when the following condition is met: if a Borel set B is such that $T^{-1}B = B$, then $\mu(B)$ is either 0 or 1. Since X is a compact metric space, (X, T) is guaranteed to have an ergodic measure [34, Prop. 3.8.9].

The Birkhoff ergodic theorem [34, Thm. 3.8.5] states that if μ is an ergodic measure, then for every integrable function h, the set of points x for which[1] $\int h d\mu \neq \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} h(T^i x)$ has μ-measure 0. Thus, for each $u \in \mathcal{L}(X)$, the set $B_u = \{x \mid f_x(u) \neq \mu([u]_X)\}$ has μ-measure 0. By Lem. 3.1.1, the complement of the countable union of all such B_u is precisely the set of points that are generic for μ. Thus, the set G of generic points for μ has μ-measure 1. If (X, T) is minimal, we additionally have that G is dense (in the topological sense) in X: indeed, G would contain the orbit of any generic point.

The exception of non-generic points does not arise in dynamical systems (X, T) that are *uniquely ergodic*, i.e., there exists only one invariant measure μ, which is guaranteed to be ergodic [34, Cor. 3.8.10]. In this case, Oxtoby's theorem [66, Thm. 4.3] tells us that all points $x \in X$ are generic for μ. Moreover, unique ergodicity is even equivalent to each word in the shift having *uniform* factor frequencies [34, Prop. 3.8.14]. A word x admits uniform factor frequencies[2]if for every $u \in \mathcal{L}(x)$ and every δ, there exists M such that for every $m \geq M$ and every $v \in \mathcal{L}_m(x)$ we have $|f_v(u) - f_x(u)| < \delta$.

Lemma 3.1.2. *If an effectively uniformly recurrent word x generates a uniquely ergodic shift X with invariant measure μ, then x admits computable factor frequencies.*

Proof. Given a factor u of length l, and a tolerance δ, we shall show that we can compute L such that for all $v \in \mathcal{L}_{L+l-1}(x)$, we have $|f_x(u) - f_v(u)| < \delta$. It then remains to count the occurrences of u in any appropriate v, e.g., $v = x(0, L + l - 1)$. For notational convenience, in this proof we abbreviate $\mathcal{L}_{L+l-1}(x)$ as simply $\mathcal{L}$.

It suffices to show, for correctness, that for every $L \geq 1$, we have $\min_{v\in\mathcal{L}} f_v(u) \leq f_x(u) \leq \max_{v\in\mathcal{L}} f_v(u)$. Indeed, for termination, by the definition of uniform frequencies, the difference between the bounds will necessarily be less than δ for some L. We can find such L by brute enumeration (made possible by the fact that x is effectively uniformly recurrent).

We prove the above required claim by showing that $f_x(u)$ is a convex combination of the values of $f_v(u)$; more precisely, we have $f_x(u) = \sum_{v\in\mathcal{L}} f_x(v) f_v(u)$. To that end,

[1] We overload inequality to also include the case where the limit does not exist.

[2] The very astute reader might notice that our definition has a slight syntactic difference from the one in [34], but semantically, the limit converged to in both definitions would be the same.

we define the continuous function χ, which maps x to $f_{x(0,L+l-1)}(u)$. We clearly have that $\sum_{v\in\mathcal{L}} f_x(v)f_v(u) = \int \chi d\mu$. It remains to show that $\lim_{N\to\infty} \frac{1}{N}\sum_{i=0}^{N-1} \chi(T^i x) = f_x(u)$, and conclude by applying unique ergodicity.

Observe that by definition, $\chi(x) = \frac{1}{L}\sum_{j=0}^{L-1} \mathbf{1}_u(T^j x)$. Thus, we can express

$$\sum_{i=0}^{N-1} \chi(T^i x) = \frac{1}{L}\sum_{i=0}^{N-1}\sum_{j=i}^{i+L-1} \mathbf{1}_u(T^j x).$$

We can further rearrange the summation to obtain

$$\sum_{i=0}^{N-1} \chi(T^i x) = \frac{1}{L}\left(L\sum_{i=L-1}^{N-1} \mathbf{1}_u(T^i x) + \sum_{i=0}^{L-2}(i+1)(\mathbf{1}_u(T^i x) + \mathbf{1}_u(T^{N+L-2-i}x))\right).$$

Asymptotically, we thus have

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=0}^{N-1} \chi(T^i x) = \lim_{N\to\infty}\frac{1}{N}\sum_{i=L-1}^{N-1} \mathbf{1}_u(T^i x).$$

The expression on the right-hand side is $(N-L+1)/N$ times the Birkhoff average of $N-L+1$ terms, and indeed converges to the limit of the usual Birkhoff averages, i.e., to $f_x(u)$, as desired. □

3.2 Unique Ergodicity is Perishable

The existence of factor frequencies is very susceptible to failure when words are drawn from minimal shift spaces that are not uniquely ergodic. We recall that the set of all invariant measures of (X,T) form a convex set whose extremal points are the ergodic measures [34, Prop. 3.8.10]. In particular, if (X,T) is not uniquely ergodic, it will have at least two distinct ergodic measures.

Lemma 3.2.1 (Thm. 1.1 in [28])**.** *Let X be a minimal shift space which is not uniquely ergodic. There exists a dense set $B \subseteq X$ of words that are not generic for any invariant measure, and hence do not admit factor frequencies.*

To summarise the discussion on the existence of factor frequencies, for a minimal shift X, we have the following. If X is uniquely ergodic, then all $x \in X$ admit (uniform and computable) factor frequencies, and moreover agree on the frequencies of each factor. On the other hand, if X is not uniquely ergodic, then it will have a dense subset of words which do not admit factor frequencies, and a dense (by virtue of including the orbit of any generic point) subset of generic points for its

ergodic measures. These generic points do admit factor frequencies, but they are not unanimous on the frequencies of all factors because they are generic for different measures.

Let us now explain how to obtain the counterexamples to the preservation of unique ergodicity alluded to in the beginning of this chapter. We follow the construction due to Chaika [21].

Lemma 3.2.2. *There exists a uniformly recurrent word $x \in \{0,1\}^\omega$ that admits factor frequencies by virtue of generating a uniquely ergodic shift, but $\mathcal{A}(x)$ does not, where $\mathcal{A}$ is the two-state automaton that changes the state upon reading* 1*, and stays in the same state upon reading* 0*.*

Proof. We shall choose x to be the symbolic trace of a carefully constructed topological dynamical system. Chaika [21] constructs a minimal and uniquely ergodic dynamical system (X', T') as follows: the space X' is the unit circle identified by the interval $[0,1)$, and T' maps ζ to $\zeta + \alpha$, where $\alpha < 1/3$ is an irrational number. The invariant measure μ' is the usual Lebesgue measure. There is a distinguished set I which is the union of the intervals $[0,\gamma)$ and $[\alpha+\beta, \alpha+\beta+\gamma)$, where β, γ are constants depending on α. The dynamics of the *skew product* $Y' = (\mathbb{Z}_2 \rtimes X', T')$ maps (b, ζ) to $(b + \mathbf{1}_I(\zeta), \zeta + \alpha)$, where $\mathbf{1}_I(\zeta)$ is 1 if $\zeta \in I$, and 0 otherwise. This skew product is minimal, but admits precisely two distinct ergodic measures ν'_0, ν'_1.

For each ζ, we define its symbolic trace $x = \mathsf{tr}(\zeta)$ as $x(n) = \mathbf{1}_I(\zeta + n\alpha)$. We claim that $\mathsf{tr}(\zeta)$ generates a minimal shift. To that end, for each $u \in \{0,1\}^+$, we shall inductively define sets I_u with the property that u occurs at index n in $\mathsf{tr}(\zeta)$ if and only if $\zeta + n\alpha \in I_u$. Clearly, $I_1 = I$, and $I_0 = [0,1) \setminus I_1$, and $I_{bu} = I_b \cap (-\alpha + I_u)$. By examining the orientation of the intervals, we have that each I_u is either empty, or contains a non-empty open set. Thus, $\mathsf{tr}(\zeta)$ is uniformly recurrent, and furthermore $\mathcal{L}(\mathsf{tr}(\zeta))$ is the set of factors u such that I_u is non-empty (we take I_ε to be $[0,1)$). It follows that every $\mathsf{tr}(\zeta)$ generates the same shift X, and that $f_{\mathsf{tr}(\zeta)}(u) = \mu'(I_u)$. The unique ergodicity of X follows readily from the following lemma.

Lemma 3.2.3. *Consider the partition of the unit circle into finitely many semi-open intervals $I_j = [\beta_j, \gamma_j)$, and let α be irrational. The shift X generated by the codings (with respect to the intervals I_j) of the rotation by α is uniquely ergodic.*

Proof. Denote by $\tilde{I}_j$ the interval $(\beta_j, \gamma_j]$. By symmetry, the set of codings with respect to the partition of "dual" intervals is also included in X. We shall prove that X comprises precisely of the codings with respect to the original partition, and codings

with respect to the dual partition. For each $u \in \mathcal{L}_n(X)$, we define I_u and $\tilde{I}_u$ as before, and also define the non-empty closed interval $\bar{I}_u$ as their union.

If $x \in X$ is a coding of the rotation starting at ζ, then for each prefix u of x, we must have $\zeta \in \bar{I}_u$. Conversely, if ζ is in each $\bar{I}_u$, then x is its coding because by the orientation of the intervals, ζ is either in each I_u, or in each $\tilde{I}_u$. (If ζ was on the left endpoint of I_u and right endpoint of $\tilde{I}_{uv}$, it would imply that I_{uv} is empty, a contradiction.)

Thus, each $x \in X$ defines an infinite sequence of nested closed intervals, and there necessarily exists a point ζ in their countable intersection. By the preceding discussion x codes a rotation of this ζ with respect to either the original partition, or the dual partition. This proves that each $x \in X$ admits factor frequencies, with $f_x(u)$ given by the Lebesgue measure of I_u. We use Oxtoby's theorem to conclude that x is uniquely ergodic. This completes the proof of Lem. 3.2.3. □

To continue with the proof of Lem. 3.2.2, We now define the shift Y to be the set of runs of $\mathcal{A}$ (starting in either state) on words of X. By Lem. 2.1.3, this shift consists of uniformly recurrent words, and is minimal (i.e., all words have the same factor language) because it is generated by symbolic traces of the minimal system $Y' = Z_2 \rtimes X'$.

Finally, we show that Y admits two distinct invariant measures ν_0, ν_1. For this, observe that $\mathsf{tr}\colon Y' \to Y$ is injective (by a simple pigeonhole argument using non-constant factor complexity) and measurable. Indeed, the subsets B of Y for which $\mathsf{tr}^{-1}(B)$ is Lebesgue-measurable form a σ-algebra which includes the cylinders of Y as their preimages are measurable sets $\{b\} \times I_u$. Thus, the ergodic measures ν'_0, ν'_1 of Y' project to invariant measures as $\nu_b = \nu'_b \circ \mathsf{tr}^{-1}$.

Also observe from the proof of Lem. 3.2.3 that tr^{-1} is defined for all but the countably many words which may only be realised as codings with respect to the dual partition: they must be initialised by points whose orbits intersect the boundary points of the intervals. The map tr^{-1} is injective and continuous where defined. In particular, if $B' \subseteq Y'$ is measurable, then so is $\mathsf{tr}(B') \subseteq Y$.

To complete the proof, we record that ν_0, ν_1 are indeed distinct because ν'_0, ν'_1 are such that each measure concentrates its mass in one "copy" of X' [21, Prop. 3]. Formally, consider disjoint sets B'_0, B'_1 that respectively have measure 1 with respect to ν'_0, ν'_1. Define the disjoint measurable sets B_0, B_1 as $\mathsf{tr}(B'_0), \mathsf{tr}(B'_1)$ respectively. By construction, $\nu_0(B_0) = \nu_1(B_1) = 1$.

We thus have that Y is a shift that is not uniquely ergodic, and contains the runs of $\mathcal{A}$ on all $x \in X$. In particular, by Lem. 3.2.1, there exists some counterexample $y = \mathcal{A}(x)$ that does not admit factor frequencies despite x doing so. □

The natural questions that follow are: can one identify when a shift is uniquely ergodic? Is there a stronger condition on X which precludes the misbehaviour of Y described above? Boshernitzan [17] gave a sufficient (but not necessary) condition in answer to the former. In this chapter, we shall show that this condition answers the latter in the affirmative.

Definition 3.2.4 (Condition (B))**.** *Let $X \subseteq \Sigma^\omega$ be a minimal shift space. For an invariant measure μ, define $\kappa\colon \mathbb{N} \to \mathbb{R}$ as*

$$\kappa(n) = \min_{u \in \mathcal{L}_n(X)} \left(n \cdot \mu\left([u]_X\right)\right).$$

The minimal shift space $X \subseteq \Sigma^\omega$ satisfies Boshernitzan's condition (also referred to as Condition (B)) if it admits an invariant measure μ such that the corresponding sequence κ satisfies $\limsup_{n\to\infty} \kappa(n) > 0$. A uniformly recurrent word $x \in \Sigma^\omega$ satisfies Boshernitzan's condition if it generates a shift space X that satisfies Boshernitzan's condition.

We notational convenience, we declare $\kappa'(n)$ to be shorthand for $\kappa(n)/n$; note that κ' is non-increasing.

Lemma 3.2.5 (Boshernitzan [17])**.** *A minimal shift space $X \subseteq \Sigma^\omega$ that satisfies Boshernitzan's condition is uniquely ergodic. If a word $x \in \Sigma^\omega$ satisfies Boshernitzan's condition, then it admits uniform factor frequencies.*

If a word x is linearly recurrent, then it satisfies Boshernitzan's condition [30, Thm. 15]. By Prop. 1.1.19, Boshernitzan's condition is also satisfied by primitive morphic words. The work of Damanik and Lenz [24] is important from the perspective of applicability: they show that Sturmian words satisfy Boshernitzan's condition, as do almost all words coding the orbits of interval-exchange transformations [25, Thm. 5], a large class of 1-dimensional toric words [25, Sec. 5], and almost all Arnoux-Rauzy words [25, Thm. 12].

We devote the remainder of this chapter to proving that if a word x satisfies Boshernitzan's condition, then so does the output $\mathcal{A}(x)$ of any transducer.

3.3 The Morphic Case

Our first step is to prove that the properties we are interested in are preserved by the action of substitutions. This allows us to focus on proving preservation theorems for automata rather than transducers. The following result extends Lem. 1.1.18, and we freely borrow the notion of attribution from its proof.

Lemma 3.3.1. *Let $\sigma\colon \Sigma \to \Gamma^*$ be a substitution, and let $x \in \Sigma^\omega$ be an infinite word. If x has one of the following properties, then $\sigma(x)$ also has that property, provided it is an infinite word.*

1. *uniformly recurrent and admits factor frequencies*
2. *satisfies Boshernitzan's condition*

In item (1), if x is effectively uniformly recurrent and admits computable factor frequencies, then $\sigma(x)$ also has these properties.

Proof. We prove item (1) in two steps. We decompose σ as $\tau \circ \hat{\sigma}$, where τ is a letter-to-letter substitution, and $\hat{\sigma}$ maps each letter a to $|\sigma(a)|$ copies of a, i.e., $\hat{\sigma}(a) = (a,0)(a,1)\cdots(a,|\sigma(a)|-1)$. We denote $\hat{\sigma}(x)$ by z, and $\tau(z)$, which is the same as $\sigma(x)$, by y. Observe by Lem. 1.1.18 that z, y are (effectively) uniformly recurrent. We shall show that z admits factor frequencies, and in turn so does y.

By the uniform recurrence of x, we have that any factor w of z can only be attributed to one of finitely many factors $\{u_1, \ldots, u_d\}$ of x, and this factor will have length at most $R_x(1) \cdot |w|$. By the definition of attribution, there is no pair u_i, u_j such that u_i is a factor of u_j; moreover by the positional-encoding nature of the construction, each u_i, upon restricting to non-erasing letters, will give a word u such that $\hat{\sigma}(u)$ contains exactly one occurrence of w. Conversely, the set of factors which begin and end with a non-erasing letter, and give u upon restricting to non-erasing letters is $\{u_1, \ldots, u_d\}$. We thus obtain $f_z(w) = f_z(\hat{\sigma}(u))$. We claim that $f_z(\hat{\sigma}(u)) = F/L$, where $F = \sum_{i=1}^{d} f_x(u_i)$, and L is the non-zero constant $\sum_{a\in\Sigma} f_x(a)|\hat{\sigma}(a)|$, which is lower-bounded by $1/R_x(1)$.

We note that

$$F = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \sum_{j=1}^{d} \mathbf{1}_{u_j}(T^i x)$$

by definition. By Lem. 3.1.1, we also have that

$$L = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} |\sigma(x(i))|$$

by expressing L as the integral of the function that maps x to the length of $\sigma(x(0))$. In order to prove that $f_z(\hat{\sigma}(u)) = F/L$, we shall rephrase the above observations by taking appropriate subsequences of Birkhoff averages.

Let $(\beta_N)_{N\in\mathbb{N}}$ denote the strictly increasing sequence whose N-th element β_N is the index (in x) of the N-th occurrence of an element of $\{u_1, \ldots, u_d\}$. We have in particular that $\lim_{N\to\infty} \frac{N}{\beta_N} = F$.

Similarly, we define the strictly increasing sequence $(\gamma_N)_{N\in\mathbb{N}}$ of indices of occurrences of $\hat{\sigma}(u)$ in z as $\gamma_N = \sum_{i=0}^{\beta_N - 1} |\sigma(x(i))|$, and observe that $\lim_{N\to\infty} \frac{\gamma_N}{\beta_N} = L$.

Since $L \neq 0$, we obtain that $\lim_{N\to\infty} \frac{N}{\gamma_N} = \frac{F}{L}$. We have almost achieved our aim of showing that

$$\lim_{M\to\infty} \frac{1}{M} \sum_{i=0}^{M-1} \mathbf{1}_{\hat{\sigma}(u)}(T^i z) = \frac{F}{L};$$

we have shown that the subsequence formed by taking $M = \gamma_N$ converges to the desired limit. However, observe that the only indices at which the original sequence of Birkhoff averages is increasing are the ones where M steps from γ_N to $\gamma_N + 1$. The corresponding increment is at most $1/\gamma_N$: thus, the limit of the full sequence is the same as that of the subsequence, i.e., $f_z(\hat{\sigma}(u)) = F/L$, as desired.

The proof that $y = \tau(z)$ inherits the property of admitting factor frequencies from z is simpler because τ is letter-to-letter. For any factor v, there are finitely many factors $w_1, \ldots, w_d$ of the same length such that $\tau(w_i) = v$. The frequency $f_y(v)$ is simply $\sum_i f_z(w_i)$.

To prove item (2),[3] observe that

$$\min_{v\in\mathcal{L}_n(y)} f_y(v) \geq \frac{1}{L} \min_{u\in\mathcal{L}_{n\cdot R}(x)} f_x(u),$$

where $R = R_x(1)$. Writing $\kappa'(n)$ as shorthand for $\kappa(n)/n$ as declared in Def. 3.2.4, using X, Y to denote the shifts generated by x, y, and defining μ_X, μ_Y using f_x, f_y, we get $L \cdot \kappa'_Y(n) \geq \kappa'_X(nR)$, and hence

$$L(R+1)n \cdot \kappa'_Y(n) \geq \max_{0\leq r<R} [(nR + r) \cdot \kappa'_X(nR + r)].$$

This allows us to conclude that $\limsup_{n\to\infty} \kappa_Y(n) > 0$, given $\limsup_{n\to\infty} \kappa_X(n) > 0$, thus establishing the preservation of Boshernitzan's condition. □

We record an interesting corollary of the above proof.

[3] The proof of [24, Thm. 8(a)], which states item (2), implicitly assumes the substitution is non-erasing.

Corollary 3.3.2. *Let σ be a non-erasing substitution. If x admits (computable) factor frequencies, then so does $\sigma(x)$.*

Regardless of how one requires primitive morphic words to be obtained from primitive substitutive words, we now have the following extension of Lem. 1.1.17

Lemma 3.3.3. *Primitive morphic words are effectively linearly recurrent and admit computable factor frequencies with closed-form expressions.*

We now show that if x is a primitive morphic word and $\mathcal{A}$ is a transducer, then $\mathcal{A}(x)$, if an infinite word, has a primitive morphic suffix.

Theorem 3.3.4. *Let $\mathcal{A} = (Q, q_{\mathsf{init}}, \Sigma, \delta, \Gamma, \lambda)$ be a transducer, and let $x \in \Sigma^\omega$ be a primitive morphic word given as the image under τ of a particular fixed point of a primitive substitution σ. When $\mathcal{A}(x)$ is an infinite word:*

1. *The word $\mathcal{A}(x)$ has a primitive morphic suffix y. We can compute the index at which y begins, and compute substitutions $\hat{\sigma}, \hat{\tau}$ such that $\hat{\sigma}$ is primitive and y is the image under $\hat{\tau}$ of a fixed point of $\hat{\sigma}$.*

2. *The word $\mathcal{A}(x)$ admits computable factor frequencies with closed-form expressions.*

Proof. We assume without losing generality that $\mathcal{A}$ is an automaton and the input x is primitive substitutive. We also observe that item (2) now follows immediately from item (1), and thus focus on proving item (1).

We first prove that $\mathcal{A}(x)$ is morphic. To that end, let us define a sequence $\mathcal{A}_0, \mathcal{A}_1, \ldots$ of automata such that $\mathcal{A}_0 = \mathcal{A}$, and $\mathcal{A}_{n+1} = (Q, q_{\mathsf{init}}, \Sigma, \delta_{n+1})$ "fast-forwards the run of $\mathcal{A}_n$ on x". Formally, $\delta_{n+1}(q, a) = \delta_n(q, \sigma(a))$. Dually to the "fast-forwarding", we define "slow-motion" substitutions $\hat{\sigma_0}, \hat{\sigma_1}, \ldots$ from $Q \times \Sigma$ to $Q \times \Sigma$ as follows. The word $\hat{\sigma_n}((q, a))$ is the run of $\mathcal{A}_n$ on $\sigma(a)$ starting in state q. By construction, we have that $\mathcal{A}_n(x) = \hat{\sigma_n}(\mathcal{A}_{n+1}(x))$.

Clearly, the set $\{\mathcal{A}_n \mid n \in \mathbb{N}\}$ is finite and can be effectively enumerated; in particular, our sequence of automata is effectively ultimately periodic. In other words, there exist an index m and a period p such that $\mathcal{A}_m$ is the same as $\mathcal{A}_{m+p}$, which induces $\mathcal{A}_n$ to be the same as $\mathcal{A}_{n+p}$ for all $n \geq m$. We thus get that $\mathcal{A}_m(x)$ is the fixed point of $\hat{\sigma_m} \cdots \hat{\sigma_{m+p-1}}$, and that $\mathcal{A}(x)$ is morphic because $\mathcal{A}(x) = \mathcal{A}_0(x) = \hat{\sigma_0} \cdots \hat{\sigma_{m-1}}(\mathcal{A}_m(x))$.

Now, since x is primitive morphic, it is effectively linearly recurrent (Lem. 3.3.3). The run $\mathcal{A}(x)$ therefore has a suffix y that is effectively linearly recurrent, and the

starting index N of this suffix can be computed (Thm. 2.1.1). This suffix y of the morphic word $\mathcal{A}(x)$ is again morphic, and the substitutions describing it can be computed (Lem. 1.1.2). Having proved that y is an effectively linearly recurrent morphic word, we invoke Durand's Prop. 1.1.19 to deduce that y is indeed primitive morphic, and we can compute substitutions $\hat{\sigma}, \hat{\tau}$ such that $\hat{\sigma}$ is primitive, and y is the image under $\hat{\tau}$ of a fixed point of $\hat{\sigma}$. □

For the sake of completeness, we comment on the case of *automatic* words. Recall that a word $x \in \Gamma^\omega$ is automatic [3, Chap. 5] if for some $k \geq 2$, there exists an automaton that computes $x(n)$ when given the k-ary representation of n as input: we then say that x is k-automatic. Clearly, for $a \geq 1$, we have x is k^a-automatic if and only if x is k-automatic. Equivalently, by a result due to Cobham [3, Thm. 6.3.2], a word x is k-automatic if and only if it is the image under a coding of a fixed point of a k-uniform morphism σ (for every $a \in \Sigma$, $|\sigma(a)| = k$). Note that the definition can be relaxed (using the discussion of Sec. 1.1.1) to replace the coding by a k'-uniform morphism [3, Cor. 6.8.3]. Our proof of Thm. 3.3.4 can thus be adapted to deduce the main result of [3, Sec. 6.9]:

Lemma 3.3.5. *If $\mathcal{A}$ is a uniform transducer and x is k-automatic, then $\mathcal{A}(x)$ is k-automatic.*

3.4 Preservation of Boshernitzan's Condition

In this section, we prove the following technical highlight of this chapter.

Theorem 3.4.1. *Let $\mathcal{A}$ be a transducer, and let $x \in \Sigma^\omega$ be a uniformly recurrent word that satisfies Boshernitzan's condition. We have that if $\mathcal{A}(x)$ is an infinite word, then its uniformly recurrent suffix also satisfies Boshernitzan's condition. Furthermore, if x is effectively uniformly recurrent, then we can compute the factor frequencies of $\mathcal{A}(x)$ too.*

The existence and effective identification of a uniformly recurrent suffix of $\mathcal{A}(x)$ have already been covered in Thm. 2.1.1. We have also described in Lem. 3.1.2 how the computability of factor frequencies follows from effective uniform recurrence. We therefore focus on proving that the requisite asymptotic behaviour of the measure is preserved. By Lem. 3.3.1, we can also assume without losing generality that $\mathcal{A}$ is an automaton.

Our proof uses the lightweight Krohn–Rhodes theorem (Prop. 1.2.8), and proceeds by a case distinction. We first prove, in a rather mechanical way, that Boshernitzan's condition is preserved assuming the presence of reset letters. The case of permutation automata needs significantly more technical insight.

3.4.1 The Case with Reset Letters

We establish the following lemma, which is slightly stronger than what we require.

Lemma 3.4.2. *Let $\mathcal{A}$ be a permutation-reset automaton, and let $x \in \Sigma^\omega$ be a uniformly recurrent word that contains a reset letter for $\mathcal{A}$. If x has the following properties, then so does the uniformly recurrent suffix of $\mathcal{A}(x)$:*

1. *admits factor frequencies*
2. *satisfies Boshernitzan's condition*

Furthermore, these properties are effective if x is effectively uniformly recurrent.

Proof. We borrow ideas from the proof of Lem. 2.1.2, which showed that (uniform) recurrence is preserved in this setting.

Let us discuss how to determine the frequency of a recurrent factor v of $\mathcal{A}(x)$, whose projection onto Σ^+ is a recurrent factor u of x. We construct a tree whose root is u, vertices are factors u' of the form wu, leaves are factors u' that have u as a proper suffix and begin with a reset letter, and the successors of the internal nodes u' are of the form au'. The depth of the tree is thus at most $R_x(1)$. The frequency $f_y(v)$ is simply $\sum_{u'} f_x(u')$, where the summation ranges over leaves u' such that the run of the reset automaton on the word u' has v as its suffix. This proves item (1)

To establish item (2), we let y denote the uniformly recurrent suffix of $\mathcal{A}(x)$, let Y denote the shift that it generates, and let μ_Y be the measure prescribed by f_y. Similarly, let X denote the shift generated by x, and let μ_X be the measure prescribed by f_x. Using the terminology of Def. 3.2.4, observe from the above argument that $\kappa'_Y(n) \geq \kappa'_X(n + R_x(1))$. Thus, for all large enough n, we have that $3 \cdot \kappa_Y(n) \geq \max(\kappa_X(2n), \kappa_X(2n+1))$, and therefore $\limsup_{n\to\infty} \kappa_Y(n) > 0$ provided $\limsup_{n\to\infty} \kappa_X(n) > 0$. □

Incidentally, the above proof, along with the Krohn–Rhodes theorem (Thm. 1.2.7) and Lem. 3.3.1, gives the following result in the special case of counter-free transducers (i.e., transducers whose underlying automata are counter-free).

Proposition 3.4.3. *Let $\mathcal{A}$ be a counter-free transducer, and let x be a uniformly recurrent word that admits factor frequencies. If $\mathcal{A}(x)$ is an infinite word, then its uniformly recurrent suffix also has these properties. Furthermore, if x is effectively uniformly recurrent, then the properties are effective for $\mathcal{A}(x)$ too.*

3.4.2 The Case of Permutation Automata

We now prove the main case of Thm. 3.4.1 through the following lemma.

Lemma 3.4.4. *Let $\mathcal{A}$ be a permutation automaton. If the uniformly recurrent word x satisfies Boshernitzan's condition, then so does the uniformly recurrent word $y = \mathcal{A}(x)$.*

The group G generated by the transitions of the permutation automaton, and the natural onto morphism $\varphi\colon \Sigma^* \to G$ (the morphism maps a word u to the element corresponding to δ_u) will play a key role in the proof. Let X be the minimal shift defined by x, let μ_X be its unique invariant measure, and recall that by Def. 3.2.4, $\limsup_{n\to\infty} \kappa_X(n) > 0$. We define the *skew product* topological dynamical system[4] $(G \rtimes X, T)$ with the compact metric state space $\{(g, x) \mid g \in G, x \in X\}$, and update $T(g, x) = (g \cdot \varphi(x(0)), x(1, \infty))$. Observe that $T^{-1} \circ (g, x) = \{(ga^{-1}, ax) \mid ax \in X\}$, where a^{-1} is shorthand for $\varphi(a)^{-1}$.

We shall equivalently (in fact, interchangeably, since we deal with a permutation automaton) regard the skew product as a shift space Z over the alphabet $(G \times \Sigma)$. We map (g, x) to y (and also denote y by (g, x)), where $y(n) = (g \cdot \varphi(x(0, n)), x(n))$. This perspective is isomorphic and each element $y \in Z$ can naturally be viewed as the run of an automaton with states G on its projection $x \in X$. Note that $\mathcal{A}(x)$ can be obtained by simply applying a coding (letter-to-letter substitution) on y. By Lem. 2.1.3, each $y \in Z$ is uniformly recurrent (however, Z is not necessarily minimal). Given a word x whose run $(e, x) = y$ we are to study, let Y be the minimal shift defined by y: note that Y is contained in (but not necessarily equal to) Z. We use π to denote the projection of Y onto X, and π^{-1} to denote its pre-image.

Our goal is to prove that if X satisfies Boshernitzan's condition (Def. 3.2.4), then so does Y. We shall do so[5] by starting with an arbitrary invariant measure ν on Y, and using it to construct an invariant measure μ_Y on Y that satisfies $\kappa_Y(n) = c \cdot \kappa_X(n)$ for all large enough n, where c is a constant.

[4] We acknowledge that the notation T is overloaded, but nevertheless use it for clarity because, as we shall see, it performs the same shift operation in spirit. If the shift operator is invoked on multiple shift spaces within the same context, we shall use a distinguishing subscript.

[5] Some proof ideas are inspired by [15], especially Prop. 3.8 *ibidem*.

We shall study the symmetry of Y in order to understand how much structure Y inherits from X. Consider the group action of G on Z, where $h \in G$ maps (g, x) to (hg, x). By the associativity of the binary group operation, we have that the action of h commutes with the shift operation T_Z and its inverse. Define the subgroup H as $\{h \in G \mid hY = Y\}$. We shall show the following connection between invariant measures on Y and X.

Lemma 3.4.5. *Let ν be an arbitrary invariant measure on Y.*

1. *The map $\nu \circ h$ is an invariant measure on Y for each $h \in H$.*
2. *The map $\nu \circ \pi^{-1}$ on X is the same as the unique invariant measure μ_X.*

In particular, the invariant measure $\mu_Y := \frac{1}{|H|} \sum_{h \in H} \nu \circ h$ satisfies $\mu_Y \circ \pi^{-1} = \mu_X$.

The following lemma, which is inspired by [51, Prop. 2.1], is instrumental in making π^{-1} explicit: it tells us that we can express $\pi^{-1}(x) = \{(g, x) \mid g \in \alpha(x)\}$.

Lemma 3.4.6. *The map α given by $\alpha(x) = \{g \mid (g, x) \in Y\}$ is a well defined function from X to $\mathcal{C} = H\backslash G$ (i.e., the set of right cosets of H), satisfies $\alpha(ax) = \alpha(x) \cdot a^{-1}$, and is continuous. In particular, X can be expressed as a finite union of cylinders on which α is constant.*

Proof of Lem. 3.4.6. Note that $\pi(Y)$ must be a shift contained in X because the projection π commutes with T and is a closed map as G is compact; since X is a minimal shift, $\pi(Y) = X$. This assures us that α maps each $x \in X$ to a nonempty set.

We first prove that $\alpha(x)$ is indeed a coset. By the defining property of the subgroup H, we deduce that if $(g, x) \in Y$, then $Hg \subseteq \alpha(x)$. We now need to prove that $\alpha(x)$ cannot span more than one coset, i.e., if $(g, x) \in Y$ and $(g', x) \in Y$, then $g' = hg$ for some $h \in H$. Let $Y' = g'g^{-1}Y$, and observe that Y' is also a minimal shift (recall that the group action commutes with the shift operation). We have that $(g', x) = (g'g^{-1}g, x)$ is contained in the intersection of the minimal shifts Y, Y': this is only possible if $Y = Y'$. In other words, $h = g'g^{-1} \in H$. We henceforth refer to α as a function from X to $\mathcal{C}$; in technical parlance it is a *(minimal) cobounding map modulo H*.

The next requirement follows readily. Let $x, ax \in X$. Since $\alpha(x)$ and $\alpha(ax)$ are both cosets, they have the same cardinality, and it suffices to observe that by construction, $\alpha(ax) \cdot \varphi(a) \subseteq \alpha(x)$ in order to establish that $\alpha(ax) = \alpha(x) \cdot a^{-1}$.

Thirdly, we show that α is continuous. Suppose for the sake of deriving a contradiction that α is not continuous at some $x \in X$, i.e., $\alpha(x) = C$, and for every index n, there exists $x_n \in X$ such that $x(0,n) = x_n(0,n)$ but $\alpha(x_n) = C_n \neq C$. By the pigeonhole principle, we deduce that there exist a coset C', infinitely many indices n, and words x_n such that $x_n(0,n) = x(0,n)$, but $\alpha(x_n) = C' \neq C$. Consider an element $g' \in C'$, and observe that $((g', x_n))_n$ is an infinite sequence of elements in the compact space Y, and this sequence converges to (g', x). However, $g' \notin \alpha(x)$, i.e., the limit does not exist in Y: a contradiction, as desired.

It remains to show that X is a finite union of cylinders on which α is constant. Indeed by continuity, for every point $x \in X$, there is a cylinder $[u]_X$ containing x on which α is constant. These cylinders cover the space; by compactness there exists a finite subcover, as desired. This proves Lem. 3.4.6. □

Proof of Lem. 3.4.5. Lem. 3.4.6 gives the framework to deduce that: (i) the operators T_Y, T_Y^{-1} on Y give the same evaluations as their counterparts T_Z, T_Z^{-1}; (ii) the inverse image π^{-1} commutes with T and T^{-1}. These will respectively imply the two parts of the claim.

Observe that $T_Y^{-1} \circ (g, x) \subseteq T_Z^{-1} \circ (g, x) = \{(ga^{-1}, ax) \mid ax \in X\}$. On the other hand, $\alpha(ax) = \alpha(x) \cdot a^{-1}$: since $g \in \alpha(x)$, this proves by the definition of α that the set inclusion is actually an equality, i.e., $T_Y^{-1} = T_Z^{-1}$ (that $T_Y = T_Z$ on Y is obvious by the definition of Y). This proves (i). It then immediately follows from associativity that $h \circ T_Y^{-1} = T_Y^{-1} \circ h$, and hence $\nu \circ h$ is an invariant measure on Y for each $h \in H$. Indeed, $\nu \circ h \circ T_Y^{-1} = \nu \circ T_Y^{-1} \circ h = \nu \circ h$.

To show (ii), observe furthermore that

$$\{(g', ax) \mid g' \in \alpha(ax), ax \in X\} = \{(ga^{-1}, ax) \mid g \in \alpha(x), ax \in X\}.$$

The set on the left expresses $\pi^{-1} \circ T_X^{-1}$, while the set on the right expresses $T_Y^{-1} \circ \pi^{-1}$. This implies that for any invariant measure ν on Y, we have $\nu \circ \pi^{-1} \circ T_X^{-1} = \nu \circ T_Y^{-1} \circ \pi^{-1} = \nu \circ \pi^{-1}$. It remains to observe that $\nu \circ \pi^{-1}$ is indeed a Borel measure on X because the function π is Borel measurable by virtue of being continuous. Thus, $\nu \circ \pi^{-1}$ is an invariant measure on X; since (X, T_X) is uniquely ergodic, it must be the same as μ_X. □

We are now ready to complete the proof that Y inherits the property of admitting an invariant measure that satisfies Boshernitzan's condition from X. Let us examine

what μ_Y evaluates to on cylinders. For convenience, we shall denote a finite factor v as (g, u), where $v(n) = (g \cdot \varphi(u(0, n)), u(n))$ for all n. We have

$$\mu_Y([(g,u)]_Y) = \frac{1}{|H|} \sum_{h \in H} \nu \circ h([(g,u)]_Y) = \frac{1}{|H|} \sum_{g' \in Hg} \nu([(g',u)]_Y),$$

and by symmetry,

$$\mu_Y([(g,u)]_Y) = \frac{1}{|H|} \cdot \sum_{g' \in Hg} \mu_Y([(g',u)]_Y).$$

Now, by Lem. 3.4.6, for all long enough factors $u \in \mathcal{L}(X)$, we have that α is constant-valued on $[u]_X$. This implies that for u long enough, we have

$$\pi^{-1}([u]_X) = \bigcup_{g' \in \alpha([u]_X)} [(g',u)]_Y.$$

By measuring both sides with respect to μ_Y, it follows that for all $g \in \alpha([u]_X)$,

$$\mu_X([u]_X) = \mu_Y \circ \pi^{-1}([u]_X) = \sum_{g' \in \alpha([u]_X)} \mu_Y([(g',u)]_Y) = |H| \cdot \mu_Y([(g,u)]_Y)$$

This proves that for all large enough n, we have $|H| \cdot \kappa_Y(n) = \kappa_X(n)$. Thus, if X satisfies Boshernitzan's condition, so does Y.

Finally, to complete the proof of Lem. 3.4.4, we obtain $\mathcal{A}(x)$ by a simple letter-to-letter substitution that substitutes (g, a) with (q, a), where q is obtained by the action of g on q_{init}. By Lem. 3.3.1, $\mathcal{A}(x)$ satisfies Boshernitzan's condition.

3.4.3 Further Technical Remarks

An open question is to find an example of a uniformly recurrent word x (or prove one does not exist) that does not satisfy Boshernitzan's condition, but $\mathcal{A}(x)$ generates uniquely ergodic shifts for every $\mathcal{A}$.

An obvious direction for future work is to generalise the following beyond the case of primitive morphic words: if the factor frequencies of x are given by *effective closed-form* expressions, then so are those of $\mathcal{A}(x)$. Concretely, consider the ability to decide whether $f_x(u) = r$ where r is a given constant: our current proof of Thm. 3.4.1 via Lem. 3.1.2 does not imply that $\mathcal{A}(x)$ inherits this property.

The main technical obstacle to the above is obtaining an effective version of Lem. 3.4.6, i.e., the computation of minimal cobounding maps on the shift X defined by the input word. This is closely related to the behaviour of *return groups*.

Given a group G and morphism φ from Σ^* into G, the return group of u (with respect to G, φ) is generated by the return words to u, and is given by $\langle \varphi(\mathcal{R}_X(u)) \rangle$.

If u' is a prefix of u, then the return group of u is a subgroup of that of u' [36, Lem. 3]. We say that the return groups *stabilise* at u if for every factor u'' such that u is a prefix of u'', the return groups of u and u'' are the same. As [15, Prop. 7.8] observes, determining minimal cobounding maps is equivalent to determining when return groups stabilise.

Our effective results for primitive morphic words indicate that the stabilisation of return groups can be determined in this case, and it has indeed been proven [36, Prop. 27]. Nevertheless, this remains open in the general case, to the best of our knowledge. A possible approach is to generalise the derivation-based techniques of Durand applied in both [36] and our paper.

Another potential approach leverages recent advances in word combinatorics, in particular the study of so-called *suffix-connected* words and their generalisations [40]. An important property of suffix-connected words (which include Sturmian words and Arnoux-Rauzy words) is that all factors are *stable* [40, Cor. 1.2], i.e., for any factor u, the return group (with respect to G, φ) is G. While this property is ideal to prove an effective version of Lem. 3.4.4, it remains difficult to determine when an automaton preserves suffix-connectedness, and hence obtain a preservation result for an entire cascade of automata. Nevertheless, as a positive case in point, we give the following application.

Proposition 3.4.7. *Let $\mathcal{A} = (Q', q_{\mathsf{init}}, \Sigma, \delta)$ be a permutation automaton, and let $x \in \Sigma^\omega$ be a Sturmian word. Let G be the group of permutations induced by the transitions, and let Q be the orbit of q_{init} under G. The run of $\mathcal{A}$ on x visits each state $q \in Q$ with frequency $1/|Q|$.*

Proof. Let X be the minimal shift defined by x. Since the word x is stable by virtue of being Sturmian and hence suffix-connected, we have by [15, Prop. 7.8] that the shift Y corresponding to the entire skew product $G \rtimes X$ is minimal. Furthermore, since x is a Sturmian word, it satisfies Boshernitzan's condition, and by Lem. 3.4.4, the shift Y also satisfies Boshernitzan's condition. From its proof, we have that the unique invariant measure on Y is given by

$$\mu_Y([(g,u)]_Y) = \frac{1}{|G|}\mu_X([u]_X).$$

Consider a state $q \in Q$. The set $H_q = \{g \mid q_{\mathsf{init}} \cdot g = q\}$ is a coset of the subgroup $H = \{h \mid q_{\mathsf{init}} \cdot h = q_{\mathsf{init}}\}$, and hence $|Q| \cdot |H| = |G|$. Using Oxtoby's theorem along with the above observations, we get that

$$f_{\mathcal{A}(x)}(q,a) = \sum_{g \in H_q} \mu_Y([(g,a)]_Y) = \frac{|H|}{|G|}\mu_X([a]_X) = \frac{1}{|Q|}f_x(a).$$

The frequency with which a run visits a state $q \in Q$ is then simply $\frac{1}{|Q|} \sum_a f_x(a) = \frac{1}{|Q|}$, as desired. □

We conclude this chapter by recording an interesting corollary of Thm. 3.4.1, and the proof of Prop. 3.4.7. We acknowledge that this fact was known to Olivier Carton and Vincent Delecroix.

Corollary 3.4.8. *Let $\mathcal{A}$ be a permutation automaton, and let $x \in \Sigma^\omega$ be a Sturmian word whose factor frequencies have effective closed-form expressions. The factor frequencies of $\mathcal{A}(x)$ also have effective closed-form expressions.*

Chapter 4

Automata on S-adic Words

In this chapter, we will show that the directive sequence of a word is an "automatic presentation" of it in the following sense: given a set of substitutions S (throughout this chapter, we assume all substitutions are non-erasing) and an ω-regular language L, we can compute an ω-regular language L' such that an S-adic word is in L if and only if its directive sequence is in L'. This result allows us, for instance (recall Sec. 1.1.3), to explicitly characterise the set of pairs of slopes and intercepts that define Sturmian words included in a given ω-regular language L; in particular we can determine whether this set is empty.

This application can be regarded as the MSO satisfiability problem for Sturmian words, and thus a counterpart in spirit to the Presburger satisfiability problem for Sturmian words studied in [45]. The results in [45] also rely on an automatic presentation of Sturmian words, which is also based on the Ostrowski numeration system, and is technically only slightly different from the directive sequence.

This chapter can also be viewed as a generalisation of the analogous results of Carton and Thomas [20] for morphic words.

4.1 Structure Lemmas for Directive Sequences

The attentive reader who recalls Lem. 1.1.4 (whose undecidability result for generated words ostensibly contradicts the program we outlined above) will immediately request that we be specific with what we mean by an "S-adic word." Indeed, we work with the notion of directed words, which is more permissive than generated words in general. We shall show how words generated by congenial expansions (recall Def. 1.1.15) form the building blocks of words that admit directive sequences. In the special case of primitive directive sequences, every directed word can also be congenially generated.

Lemma 4.1.1. *If $x \in \Sigma^+ \cup \Sigma^\omega$ is generated by a directive sequence $\boldsymbol{\sigma}$, then it is also directed by $\boldsymbol{\sigma}$.*

Proof. Let $\hat{\boldsymbol{\sigma}} = ((\sigma_n, a_n))_n$ be the augmented directive sequence that generates x. For each $m \in \mathbb{N}$, define the set $U_m = \{\sigma_m \cdots \sigma_n(a_n) \mid n \geq m\}$. By compactness, every such set U_m has an accumulation point, and we shall prove that for every such accumulation point $x_m \in \overline{U_m}$, there exists an accumulation point $x_{m+1} \in \overline{U_{m+1}}$ such that $x_m = \sigma_m(x_{m+1})$. This would suffice to prove the direction by $\boldsymbol{\sigma}$ of the only accumulation point x of U_0.

Consider the accumulation point $x_m \in \overline{U_m}$, given by $x_m = \lim_{k\to\infty} \sigma_m \cdots \sigma_{n_k}(a_{n_k})$. We may rewrite the limit as $\lim_{k\to\infty} \sigma_m(\sigma_{m+1} \cdots \sigma_{n_k}(a_{n_k}))$, i.e., the limit of σ_m applied to a sequence of words in U_{m+1}. By compactness, this sequence itself has a subsequence that converges to a limit, say $x_{m+1} \in \overline{U_{m+1}}$. By continuity of σ_m, the limit x_m of σ_m applied to this converging subsequence is equal to σ_m applied to the limit x_{m+1} of the subsequence. □

Indeed, every directive sequence congenially generates (and hence generates) some nonempty word.

Lemma 4.1.2. *Let Σ be an alphabet, and let $\boldsymbol{\sigma}$ be a directive sequence. The set of congenial sequences that augment $\boldsymbol{\sigma}$ is nonempty, and has at most $|\Sigma|$ elements.*

Proof. The fact that there is at least one congenial sequence follows from König's lemma: consider the graph whose vertices are labelled by $\Sigma \times \mathbb{N}$, and draw an edge from (b, i) to $(c, i+1)$ whenever $\sigma_{i+1}(c) = b$. By the pigeonhole principle, this infinite graph contains a finitely-branching infinite tree rooted at a vertex of the form $(a, 0)$. This tree, by König's lemma, has an infinite path, which corresponds to the desired congenial sequence.

Now we suppose, for the sake of deriving a contradiction, that there are at least $|\Sigma| + 1$ distinct ways of augmenting $\boldsymbol{\sigma}$ into a congenial sequence. Let us denote them as $(\sigma_n, a_n^{(i)})_n$ with $0 \leq i \leq |\Sigma|$. For every index n, there exist distinct i, j such that $a_n^{(i)} = a_n^{(j)}$. By applying the pigeonhole principle again, we deduce that for some pair i, j, there exist infinitely many indices n such that $a_n^{(i)} = a_n^{(j)}$. By the congeniality constraint, this in fact necessitates $a_n^{(i)} = a_n^{(j)}$ for all n, contradicting the distinction between the two congenial sequences. □

We record an important special case of the above lemma. We say that a substitution σ is *left-proper* if for all letters b, the word $\sigma(b)$ has the same starting letter.

For example, the Arnoux-Rauzy substitutions λ_i that map i to i and j to ij are left-proper. If a directive sequence has infinitely many left-proper terms, then it can be augmented into only one congenial sequence. Recall from Sec. 1.1.3 that for Sturmian words, if a directive sequence corresponds to the Ostrowski numeration, then it has infinitely many left-proper terms.

We also observe that if the directive sequence is primitive, then the words generated by the augmenting congenial sequences will be in Σ^ω. Thus, a word is unambiguously specified by a primitive directive sequence with infinitely many left-proper terms, e.g., Sturmian words and Ostrowski numeration-based directive sequences.

Given a directive sequence $\boldsymbol{\sigma}$, let $\mathsf{congenials}(\boldsymbol{\sigma})$ denote the set of words in $\Sigma^+ \cup \Sigma^\omega$ that are generated by some congenial sequence that augments $\boldsymbol{\sigma}$.

Lemma 4.1.3. *Let $\boldsymbol{\sigma}$ be a directive sequence. We have that $x \in \Sigma^+ \cup \Sigma^\omega$ is directed by $\boldsymbol{\sigma}$ if and only if x is a (possibly infinite) concatenation of words in* $\mathsf{congenials}(\boldsymbol{\sigma})$.

Proof. Let $x = x^{(0)} = u_0 u_1 \cdots$, where we have $u_n = u_{n,0} = \lim_{m\to\infty} \sigma_0 \cdots \sigma_m(a_{n,m})$ for each n. Define $u_{n,m} = \lim_{k\to\infty} \sigma_{m+1} \cdots \sigma_k(a_{n,k})$, and define $x^{(m)} = u_{0,m} u_{1,m} \cdots$. We have by construction that $x^{(m)} = \sigma_m(x^{(m+1)})$, proving the "if" part of the statement.

To prove the converse, we will show that x is either congenially generated itself, or is the concatenation of a congenially generated prefix u with a suffix y directed by $\boldsymbol{\sigma}$. Indeed, consider the sequence of words $x^{(n)}$ defined by the directive sequence, and define a sequence $a_0, a_1, \ldots$ of letters by taking a_n to be the starting letter of $x^{(n+1)}$. The sequence $((\sigma_n, a_n))_n$ is congenial by construction, and generates either x itself, or a finite prefix u of x.

In the latter case, the nondecreasing sequence $|\sigma_0 \cdots \sigma_n(a_n)|$ becomes eventually constant; let it be so from index m onwards. In other words, at sufficiently high indices, we have $\sigma_n(a_n) = a_{n-1}$. Define $u^{(n)} = x^{(n)}(0)$ for $n > m$, and $u^{(n)} = \sigma_n \cdots \sigma_m(a_m)$ otherwise. Define $y^{(n)}$ such that $x^{(n)} = u^{(n)} y^{(n)}$.

By premise, we have that $x^{(n)} = \sigma_n(x^{(n+1)})$, and by the preceding construction, we have that $u^{(n)} = \sigma_n(u^{(n+1)})$. This necessitates $y^{(n)} = \sigma_n(y^{(n+1)})$, as desired. □

Remark 4.1.4. *One may be motivated to address shifts of directed (or congenially generated) words using S-adic sequences. This can be done using* centred representations *(see, e.g., [16]). The idea is to augment directive sequences over S into sequences over $S \times \Sigma \times \mathbb{N}$, where each term (σ_n, a_n, k_n) has the property that $0 \le k_n < |\sigma_n(a_n)|$, and a_{n-1} is the k_n-th letter of $\sigma_n(a_n)$. We then direct the word*

$\lim T^{k_0}\sigma_0 T^{k_1}\sigma_1 \cdots T^{k_n}\sigma_n(a_n)$. *This may seem reminiscent of Bratteli-Vershik odometers or Dumont-Thomas numeration systems; in fact under some precise* recognisability *conditions (see, e.g., [16, Sec. 6]), these perspectives are isomorphic.*

4.2 Congeniality and Congruence

In view of Lem. 4.1.3, let us first understand how to decide, given an ω-regular language L and a congenial sequence $\hat{\boldsymbol{\sigma}} = ((\sigma_n, a_n))_n$, whether the word x generated by $\hat{\boldsymbol{\sigma}}$ is in L. Recall from Lem. 1.1.16 that

$$x = \lim_{n\to\infty} \sigma_0(a_0) \cdot \sigma_0(v_1) \cdot \sigma_0\sigma_1(v_2) \cdots (\sigma_0 \cdots \sigma_{n-1}(v_n)),$$

where $\sigma_n(a_n) = a_{n-1}v_n$. This expression prompts us to view the language L from an algebraic perspective. Let L be recognised by a morphism h into a finite ω-semigroup $M = (M_f, M_\omega)$. We have $x \in L$ if and only if $h(x) \in F$, where $F \subseteq M_\omega$ is a set of accepting elements. We thus need to determine whether

$$h \circ \sigma_0(a_0) \cdot \prod_{n=1}^{\infty} h \circ \sigma_0 \cdots \sigma_{n-1}(v_n) \in F. \tag{4.1}$$

In this section, we show that each term of the product can be computed by an automaton, and thus by Lem. 1.2.6, we can also use an automaton to determine whether the overall product is an element of F.

We define the following three functions, each with finite image. In what follows, $\mathsf{morphisms}_M$ denotes the finite set of morphisms from Σ^+ into M_f.

- $\mathsf{head} : S(\Sigma) \to (\Sigma \to \Sigma)$ maps (σ, a) to $\sigma(a)(0)$
- $\mathsf{compose} : S(\Sigma) \to (\mathsf{morphisms}_M \to \mathsf{morphisms}_M)$ maps (σ, g) to $g \circ \sigma$
- $\mathsf{tail} : S(\Sigma) \to (\mathsf{morphisms}_M \times \Sigma \to M_f \cup \{1_M\})$ maps (σ, g, a) to $g(v)$, where $1_M = g(\varepsilon)$ is a fresh neutral element that extends the semigroup M_f into a monoid, and $\sigma(a) = \mathsf{head}(\sigma, a) \cdot v$

The product in eq. (4.1) is then expressed as

$$h \circ \sigma_0(a_0) \cdot \prod_{n=1}^{\infty} \mathsf{tail}(\sigma_n, \mathsf{compose}(\sigma_0 \cdots \sigma_{n-1}, h), a_n).$$

We say that substitutions σ, τ are equivalent (modulo M) if they agree on the three functions defined above. We denote this as $\sigma \equiv_M \tau$, and use Ξ_M to denote the set of equivalence classes. Define $\mathsf{trace}_M : S(\Sigma) \to \Xi_M$ to return the equivalence class of a

substitution, and extend trace to return words over Ξ_M and $\Xi_M \times \Sigma$ respectively for sequences over $S(\Sigma)$ and $S(\Sigma) \times \Sigma$.

We will now observe that this equivalence is in fact a congruence, i.e., if $\sigma_1 \equiv_M \tau_1$ and $\sigma_2 \equiv_M \tau_2$ then $\sigma_1\sigma_2 \equiv_M \tau_1\tau_2$. Indeed, this follows readily from the following.

$$\mathsf{head}(\sigma_1\sigma_2, a) = \mathsf{head}(\sigma_1, \mathsf{head}(\sigma_2, a)) \tag{4.2}$$

$$\mathsf{compose}(\sigma_1\sigma_2, g) = \mathsf{compose}(\sigma_2, \mathsf{compose}(\sigma_1, g)) \tag{4.3}$$

$$\mathsf{tail}(\sigma_1\sigma_2, g, a) = \mathsf{tail}(\sigma_1, g, \mathsf{head}(\sigma_2, a)) \cdot \mathsf{tail}(\sigma_2, \mathsf{compose}(\sigma_1, g), a). \tag{4.4}$$

In the lemma below, $\pi(y)$ denotes the product of the semigroup elements that are the letters of the word y (i.e., the letter 1_M is excluded when taking the product).

Lemma 4.2.1. *Fix an alphabet Σ. Let $M = (M_f, M_\omega)$ be an ω-semigroup, and let h be a morphism from Σ^+ into M_f. We can construct a deterministic transducer $\mathcal{A}$ with input alphabet $\Xi_M \times \Sigma$ and output alphabet $M_f \cup \{1_M\}$ to have the following property. For any congenial sequence $\hat{\boldsymbol{\sigma}}$ that generates the word x, the transducer output $y = \mathcal{A}(\mathsf{trace}(\hat{\boldsymbol{\sigma}}))$ satisfies $\pi(y) = h(x)$.*

The required transducer uses its finitely many states to track the equivalence class of $\sigma_0 \cdots \sigma_n$.

Applying Lem. 1.2.6, we get the following main result for congenial sequences. (It is straightforward to check the local condition of congeniality with an automaton that has access to the function head.)

Theorem 4.2.2. *Fix an alphabet Σ. Let $L \subseteq \Sigma^+ \cup \Sigma^\omega$ be the union of a regular language and a ω-regular language (recognised by ω-semigroup M).*

1. *We can construct an automaton $\mathcal{A}$ over the alphabet $\Xi_M \times \Sigma$ such that for any sequence $\hat{\boldsymbol{\sigma}}$, the automaton $\mathcal{A}$ accepts $\mathsf{trace}(\hat{\boldsymbol{\sigma}})$ if and only if $\hat{\boldsymbol{\sigma}}$ is congenial and generates a word $x \in L$.*

2. *As a special case, when we have a finite set $S \subseteq S(\Sigma)$, we can construct an ω-regular language $L' \subseteq (S \times \Sigma)^\omega$ such that $\hat{\boldsymbol{\sigma}} \in L'$ if and only if it is a congenial sequence that generates a word $x \in L$.*

Since ω-regular languages are closed under projection, we can immediately record the following corollary.

Corollary 4.2.3. *Fix an alphabet Σ. Let $L \subseteq \Sigma^+ \cup \Sigma^\omega$ be the union of a regular language and a ω-regular language (recognised by ω-semigroup M).*

1. *We can construct an automaton $\mathcal{A}$ over the alphabet Ξ_M such that the automaton $\mathcal{A}$ accepts* $\mathsf{trace}(\boldsymbol{\sigma})$ *if and only if* $\mathsf{congenials}(\boldsymbol{\sigma}) \cap L \neq \{\}$.

2. *As a special case, when we have a finite set $S \subseteq S(\Sigma)$, we can construct an ω-regular language $L' \subseteq S^\omega$ such that $\boldsymbol{\sigma} \in L'$ if and only if* $\mathsf{congenials}(\boldsymbol{\sigma}) \cap L \neq \{\}$.

We would also like to prove a version of the above result where we replace "word generated congenially by $\boldsymbol{\sigma}$" with "word directed by $\boldsymbol{\sigma}$". We use Lem. 4.1.3 (a directed word is a concatenation of congenial words) in order to do so.

Let the ω-semigroup $M = (M_f, M_\omega)$, morphism h, and the set $F \subseteq M_f \cup M_\omega$ of accepting elements be as before. We construct sets $G_1, G_2, \ldots, G_d \subseteq M_f \cup M_\omega$ such that for each i, the semigroup generated by G_i intersects F. We then use Cor. 4.2.3 to define corresponding languages $L'_1, L'_2, \ldots, L'_d$ such that L'_i is the set of (traces of) directive sequences that, for each $m \in G_i$, congenially generate a word x_m such that $h(x_m) = m$. Our required language L' is then the union of $L'_1, L'_2, \ldots, L'_d$.

Theorem 4.2.4. *Fix an alphabet Σ. Let $L \subseteq \Sigma^+ \cup \Sigma^\omega$ be the union of a regular language and a ω-regular language (recognised by ω-semigroup M).*

1. *We can construct an automaton $\mathcal{A}$ over the alphabet Ξ_M such that the automaton $\mathcal{A}$ accepts* $\mathsf{trace}(\boldsymbol{\sigma})$ *if and only if $\boldsymbol{\sigma}$ directs a word $x \in L$.*

2. *As a special case, when we have a finite set $S \subseteq S(\Sigma)$, we can construct an ω-regular language $L' \subseteq S^\omega$ such that $\boldsymbol{\sigma} \in L'$ if and only if $\boldsymbol{\sigma}$ directs a word $x \in L$.*

4.3 Partial Quotients

In this section, we refine Thm. 4.2.4 for the class of Sturmian words. These words share the same factor-complexity function $n \mapsto n + 1$, and have directive sequences intimately connected to the Ostrowski representation of their slopes and intercepts. An analogue of our result may also be stated for d-letter Arnoux-Rauzy words: they have the common factor-complexity function $n \mapsto (d-1)n + 1$, and each word has a "canonical" primitive left-proper directive sequence [38, Thm. 4.12].

Lemma 4.3.1. *Let $\Sigma = \{0, 1, \ldots, d-1\}$ be the d-letter alphabet, and let $S = \{\lambda_0, \rho_0, \ldots, \lambda_{d-1}, \rho_{d-1}\} = \Lambda \cup \bar{\Lambda}$ be the set of Arnoux-Rauzy substitutions. Every Arnoux-Rauzy word $x \in \Sigma^\omega$ has a directive sequence $\boldsymbol{\sigma} \in S^\omega$ that is: (i) primitive, i.e., for each i, either λ_i or ρ_i appears infinitely often, (ii) left-proper, i.e., there*

are infinitely many letters from Λ, and (iii) normalised, i.e., factors in the language $\rho_i(\bar{\Lambda})^*\lambda_i$ are forbidden for each i.

It is easy to check that the set of primitive, left-proper and normalised directive sequences is an ω-regular subset of S^ω. We mention that there is recent work [37] that describes the set of (traces of) valid directive sequences of *dendric words*, a combinatorial generalisation of Arnoux-Rauzy words, as an ω-regular language.

We briefly digress to make an interesting remark. In general, if we want to prove that the set of (traces of) primitive directive sequences is ω-regular, we must further refine the congruence Ξ_M by agreement on the function $\mathsf{introduces} : S(\Sigma) \to (\Sigma \to \mathcal{P}(\Sigma))$, which maps (σ, a) to the set of letters in $\sigma(a)$. Clearly,

$$\mathsf{introduces}(\sigma_1\sigma_2, a) = \bigcup_{b \in \mathsf{introduces}(\sigma_2, a)} \mathsf{introduces}(\sigma_1, b).$$

It is easy to construct a nondeterministic Büchi automaton that accepts the traces of directive sequences that are *not* primitive: this automaton guesses an offending index m and checks that there is no r such that $\sigma_m \cdots \sigma_{m+r}$ is primitive. This suffices to prove that the set of (traces of) primitive directive sequences is ω-regular. We now return from our digression to our main exposition of partial quotients.

Observe that expression (1.8) (reproduced below for convenience) factorises the directive sequence of a Sturmian word into blocks τ_n witness its primitivity and left propriety. (By the rules of Ostrowski expansion, we cannot have $a = b$ at consecutive indices.)

$$\tau_n = \lambda_0^{a_{2n+1}-b_{2n+1}} \rho_0^{b_{2n+1}} \lambda_1^{a_{2n+2}-b_{2n+2}} \rho_1^{b_{2n+2}} = \tau_{n,0}\tau_{n,1}.$$

We refer to τ_n as a *partial quotient*. We can also analogously define partial quotients for Arnoux-Rauzy directive sequences by a greedy factorisation.

Theorem 4.3.2. *Let $L \subseteq \{0,1\}^\omega$ be an ω-regular language recognised by the ω-semigroup M. We can compute N such that for any Sturmian word x with slope η and intercept χ, the first N terms of the continued fraction expansion of η and the first N terms of the base-η Ostrowski expansion of χ determine whether $x \in L$.*

Proof. We begin by observing that if x is directed by $\boldsymbol{\sigma} = \sigma_0\sigma_1\cdots$, we can factorise the directive sequence into partial quotients and obtain a new directive sequence $\boldsymbol{\tau} = \tau_{0,0}\tau_{0,1}\tau_{1,0}\tau_{1,1}\cdots$, where each $\tau_{i,0}\tau_{i,1}$ is primitive and left-proper, and if $\tau_{n,b}$ is not left-proper, then the substitution $\tau_{n',b'}$ immediately before it it is necessarily of the form λ_{1-b}^k.

We record the trace of $\boldsymbol{\tau}$ using the alphabet $\Gamma = \Xi_M \times \{l, r, o\} \times \{0, 1\}$, where the first component records the congruence class of $\tau_{n,b}$, the second records whether this substitution is of the form λ_b^k, or ρ_b^k, or neither, and the third component records whether we are representing a substitution in $\lambda_0^*\rho_0^*$ or $\lambda_1^*\rho_1^*$. By the rules of Ostrowski expansion, the set Ω of valid traces is closed, and thus a compact subset of Γ^ω.

For each $\gamma \in \Gamma$, we can decide whether there exists a sequence of substitutions whose class is γ, and if it does, compute the composition π of one such sequence. By Thm. 4.2.4, we have that $x \in L$ if and only if the unique word directed by $\pi_{0,0}\pi_{0,1}\pi_{1,0}\pi_{1,1}\cdots$ is in L, where $\pi_{n,b}$ is the representative of $\tau_{n,b}$.

Now, a prefix of this directive sequence of substitutions $\pi_{n,b}$ defines a prefix of some Sturmian word y. If we take the prefix of $\boldsymbol{\pi}$ to be long enough, it will be *saturated* in the sense of Cor. 2.1.6, i.e., the defined prefix of y contains enough information to determine whether $y \in L$, and any Sturmian word y' that has this prefix in common with y will agree with y upon membership in L. We can naturally extend the definition of saturation to prefixes of words in $\Omega \subseteq \Gamma^\omega$: a prefix is saturated when its representative is.

Finally, we argue that there exists N such that a saturated prefix of a word in Ω has length at most N. Indeed, by Cor. 2.1.6, every word in Ω has a saturated prefix, and for all M, the set of words that have a saturated prefix of length M is open. These sets thus form an open cover of the compact set Ω, which must admit a finite subcover; we take N to be the maximal M in this subcover. □

4.4 Periodic Generating Sequences

In this section, we will consider a language $L \subseteq \Sigma^+ \cup \Sigma^\omega$ recognised by an ω-semigroup M, and describe pairs $(\sigma, u) \in S(\Sigma) \times \Sigma^*$ such that $\sigma^\omega(u) \in L$, where $\sigma^\omega(u) = \lim_{n\to\infty} \sigma^n(u)$.

In order to do so, we refine $\equiv_M$ to relate two substitutions if and only if they agree on the function

$$\mathsf{segments} : S(\Sigma) \to (\mathsf{morphisms}_M \times \Sigma \to \mathsf{lists}(\Sigma \times (M_f \cup \{1_M\}))),$$

which maps (σ, g, a) to the list $\langle (b_1, m_1), \ldots (b_k, m_k)\rangle$ that has the following property. The word $\sigma(a)$, when factorised based on the first occurrences of letters, is of the form $b_1 v_1 \cdots b_k v_k$, and for each i, we have $g(v_i) = m_i$. (Note that $v_i \in \{b_1, \ldots, b_i\}^*$.) This ensures that segments has finite image. Observe that agreement

on segments implies agreement on all the functions we have mentioned before, namely, head, tail, compose, introduces.

To show that $\equiv_M$ remains a congruence, we will check that we can compute $\mathsf{segments}(\sigma_1\sigma_2)$ using only $\mathsf{segments}(\sigma_1)$ and $\mathsf{segments}(\sigma_2)$. Let us describe the computation of $\mathsf{segments}(\sigma_1\sigma_2, g, a)$. First, we denote $\sigma_2(a)$ by $b_1v_1 \cdots b_kv_k$. Next, we denote $\sigma_1\sigma_2(a)$ by

$$c_{1,1}w_{1,1} \cdots c_{1,l_1}w_{1,l_1}\sigma_1(v_1) \cdots c_{k,1}w_{k,1} \cdots c_{k,l_k}w_{k,l_k}\sigma_1(v_k).$$

By construction, the first occurrence of any letter in the word $\sigma_1\sigma_2(a)$ must be among the $c_{i,j}$. We determine $m_{i,j} = g(w_{i,j})$ from the j-th element of segments applied to the substitution σ_1, the morphism g, and the letter from the i-th element from the output of $\mathsf{segments}(\sigma_2, g, a)$. We determine $n_i = g \circ \sigma_1(v_i)$ from the i-th element of $\mathsf{segments}(\sigma_2, \mathsf{compose}(\sigma_1, g), a)$. We now compile our list from the sequence

$$c_{1,1}m_{1,1} \cdots c_{1,l_1}m_{1,l_1}n_1 \cdots c_{k,1}m_{k,1} \cdots c_{k,l_1}m_{k,l_1}n_k \in (\Sigma \cup M_f \cup \{1_M\})^+$$

by iteratively following two rules:

- Whenever there are two consecutive monoid elements, we replace them by their product.
- Whenever there is a letter $c \in \Sigma$ that has occured previously, we replace it by $g(c)$.

This establishes that the refined $\equiv_M$ continues to be a congruence, and as before, we use Ξ_M to denote the finite set of congruence classes.

Our main result of this section is as follows. We devote the rest of the section to proving it.

Theorem 4.4.1. *Let $L \subseteq \Sigma^+ \cup \Sigma^\omega$ be recognised by a morphism h into a finite ω-semigroup M with accepting elements F. For any word $u \in \Sigma^+$, we can compute $G_u \subseteq \Xi_M$ such that the word $\sigma^\omega(u)$ is well defined and in L if and only if the congruence class of σ is in G_u.*

Proof. To prove the theorem, we will show how to compute $h \circ \sigma^\omega(u)$ using only the congruence class ξ of σ. We shall argue that we can do so by computing $h \circ \sigma^\omega(a)$ for every letter a of u till we reach the end of the word, or find a letter for which $\sigma^\omega(a)$ is an infinite word, or is undefined.

We first observe that $\sigma^\omega(u) = \sigma^\omega(\sigma(u))$, since $\lim_{n\to\infty} \sigma^n(u) = \lim_{n\to\infty} \sigma^{n+1}(u)$. We also observe that the map σ^ω is distributive in the following sense. Consider $\sigma^\omega(uv)$. If $\sigma^\omega(u)$ is an infinite word x, or undefined, then so is $\sigma^\omega(uv)$. If $\sigma^\omega(u)$ is a finite word, then $\sigma^\omega(uv) = \sigma^\omega(u) \cdot \sigma^\omega(v)$.

In fact, for any letter a, if $\sigma^\omega(a)$ is a finite word w, then for all $n \geq |\Sigma|$, we have $\sigma^n(a) = w$. This is because we can inductively construct sets $\Sigma_0, \Sigma_1, \ldots$ of letters such that Σ_0 is the set of letters a such that $\sigma(a) = a$, and Σ_{n+1} is the set of letters a such that $\sigma(a) \in \Sigma_n^+$. A fixed point will be reached in at most $|\Sigma|$ iterations. We can compute the set Σ_0 as letters a for which $\mathsf{tail}(\sigma, a) = 1_M$ and $\mathsf{head}(\sigma, a) = a$. We compute Σ_{n+1} as letters a for which $\mathsf{introduces}(\sigma, a) \subseteq A_n$. We thus compute $\Sigma_{|\Sigma|}$, and for each $a \in \Sigma_{|\Sigma|}$, we can determine $h \circ \sigma^\omega(a)$ as $g(a)$, where $g = \mathsf{compose}(h, \sigma^{|\Sigma|})$.

Let us describe the set of letters a for which $\sigma^\omega(a)$ is undefined. Consider a graph whose vertices correspond to letters, and there is an edge from a to b if $b = \mathsf{head}(\sigma, a)$. Clearly, for any letter a corresponding to a vertex along a path that leads to a nontrivial cycle, we have that $\sigma^\omega(a)$ is undefined because the starting letter of $\sigma^n(a)$ keeps alternating. Let Γ_0 be the set of such letters. Let Γ_{n+1} be the set of letters a such that $\sigma(a) \in \Sigma_{|\Sigma|}^* \Gamma_n \Sigma^*$. This inductive construction will saturate in at most $|\Sigma|$ iterations; for all $a \in \Gamma_{|\Sigma|}$ we have that $\sigma^\omega(a)$ is undefined. We can compute $\Gamma_{|\Sigma|}$ using the function $\mathsf{segments}$.

Let Θ denote $\Sigma \setminus (\Gamma_{|\Sigma|} \cup \Sigma_{|\Sigma|})$. For any $a \in \Theta$, we necessarily have $\sigma(a) \in \Sigma_{|\Sigma|}^* \Theta \Sigma^*$. If $\sigma(a) \in a\Sigma^+$, then we are back in the congenial setting (in fact, in the classic setting of Carton and Thomas [20]) and we compute $h \circ \sigma^\omega(a)$ as the ultimately periodic infinite product

$$a \cdot \mathsf{tail}(\sigma, h) \cdot \prod_{n=1}^{\infty} \mathsf{tail}(\sigma, \mathsf{compose}(\sigma^n, h), a).$$

If $\sigma^j(a) \in \Sigma_{|\Sigma|}^+ a \Sigma^*$, i.e., $\sigma^j(a) = uav$ for some $j \leq |\Sigma|$, we have that $\sigma^\omega(a) = (\sigma^\omega(u))^\omega$, and $h \circ \sigma^\omega(a) = g(u)^\omega$, where $g(u)$ can be determined by inspection from $\mathsf{segments}(\sigma, \mathsf{compose}(\sigma^{|\Sigma|}, h), a)$.

Now by a straightforward pigeonhole argument on the first letter from Θ in the image of $\sigma^k(a)$, for arbitrary $a \in \Theta$, there exists an iterate i (which can be computed using the congruence class of σ) such that either (A) $\sigma^i(a) = ubw \in \Sigma_{|\Sigma|}^* b \Sigma^*$ with $b \in \Theta$ and $\mathsf{head}(\sigma, b) = b$, or (B) $\sigma^i(a) = ubw_1 \in \Sigma_{|\Sigma|}^* b \Sigma^*$ with $b \in \Theta$ and there exists j such that $\sigma^j(b) = vbw_2 \in \Sigma_{|\Sigma|}^+ b \Sigma^*$. We then have that $\sigma^\omega(a) = \sigma^\omega(u)\sigma^\omega(b)$, where the image of the latter factor is computed according to the two special cases above, and the image of the former factor can be elicited from appropriate invocations of $\mathsf{segments}$.

In this manner, we can implement a dynamic-programming approach to determine $h \circ \sigma^\omega(u)$, using only the congruence class χ of σ. The set G_u consists of those congruence classes for which $h \circ \sigma^\omega(u) \in F$. □

4.5 Perspective and Future Work

During a recent workshop[1], Laurent Bartholdi announced a forthcoming line of work in collaboration with Ivan Mitrofanov. They study the first-order theory of a universe of S-adic words, where directive sequences give automatic representations of elements of the universe in a way that is amenable to the implementation of various relations. (By "automatic representation", we mean a surjection from an ω-regular language onto the universe. By "implementation", we mean that the relations are defined by ω-regular languages over the appropriate product alphabet; in particular we want to implement the equality relation.) In this section, we record what we now know already when we take the universe to be the set of Sturmian words. In this case, we can refine [45, Thm. A].

As usual, we let the slope $\eta = [0; 1 + a_1, a_2, a_3, \ldots]$, and let the digits of the base-η Ostrowski expansion of the (formal) intercept χ be $(b_1, b_2, b_3, \ldots)$. We generate the Sturmian word x with these parameters by its canonical directive sequence

$$\lambda_0^{a_1-b_1}\rho_0^{b_1}\lambda_1^{a_2-b_2}\rho_1^{b_2}\lambda_0^{a_3-b_3}\rho_0^{b_3}\cdots .$$

Following [45], the word x is represented by a sequence z over the alphabet $\Gamma \times \Gamma \times \Gamma$, where $\Gamma = \{0, 1, \#\}$. The three components of z are denoted z_η, z_χ, and $z_{-\chi}$, and respectively serve to encode the continued fraction quotients, formal intercept, and a technically convenient checksum. These component words are *aligned* in the sense that for all indices n, they agree upon whether the n-th letter is the delimiter $\#$. The words are thus divided into blocks: $z_\eta = u_0\#u_1\#\cdots$, $z_\chi = v_0\#v_1\#\cdots$, and $z_{-\chi} = w_0\#w_1\#\cdots$.

Consider $u_n = u_n(0)\cdots u_n(l)$, where $|u_n| = l + 1$. This word encodes the partial quotient $a_n = \sum_{i=0}^{\ell} u_n(i) \cdot 2^i$. We maintain that $u_n(l-1) = 1$, and $u_n(l) = 0$. In a similar manner, v_n encodes the term b_n of the formal intercept, and w_n encodes $a_n - b_n$. It is straightforward to check that the set of valid representations is an ω-regular language.

Clearly, we can implement the equality relation, and the relation SameShift that checks whether words x and y are in the same shift (by virtue of having the same

[1] I am grateful to Joël Ouaknine, Valérie Berthé, and Florian Luca for their organisation.

slope). By Lem. 1.1.6, we can also implement unary predicates $\mathsf{LowMech}, \mathsf{UpMech}$ that respectively include all lower mechanical words, and all upper mechanical words. (Recall that unless $\chi \in \mathbb{Z} + \mathbb{Z}_{\leq -1}\eta$, a Sturmian word is both upper and lower mechanical.)

In the same spirit, we can implement a unary predicate Char that indicates whether a word x is characteristic Sturmian (v_n encodes 0 for all n), and also a predicate $\mathsf{CharOrbit}$ that indicates whether a word x is in the shift orbit of the characteristic Sturmian word (by [45, Fact 2.8], the block v_n eventually encodes 0 for all n). One may already observe that this structure embeds the natural numbers $\mathbb{N}$. We can indeed do more.

It has been proven that the properties of the Ostrowski enumeration system also allow us to implement addition [45, Sec. 4], and that the representation we just introduced is amenable to *uniform* Ostrowski addition, i.e., the same automaton is capable of performing addition in any base η. We can thus implement a relation $\mathsf{plus}(.,.,.)$, which is the set of triples x_1, x_2, x_3 of Sturmian words with the same slopes η, all lower mechanical or all upper mechanical, and intercepts χ_1, χ_2, χ_3 related as $\chi_3 \equiv \chi_1 + \chi_2 \bmod 1$ (see [45, Lem. 4.5]).

The predicates $\mathsf{Char}, \mathsf{CharOrbit}, \mathsf{plus}$ together embed Presburger arithmetic into our structure, and also yield the shift relation $\mathsf{Shift} = \{(x, x') \mid x' = Tx\}$ and the orbit relation $\mathsf{Orbit} = \{(x, x') \mid \exists k \in \mathbb{N}.\ x' = T^k x\}$. When we want to use the number 0, we declare it as $\exists y_0.\ \mathsf{Char}(y_0)$. Indeed, we declare a variable $p \in \mathbb{N}$ as $\exists y_p.\ \mathsf{CharOrbit}(y_p)$, and we declare $i \leq p$ as $\exists y.\ \mathsf{CharOrbit}(y) \wedge \mathsf{plus}(y_i, y, y_p)$. We can then declare arbitrary constants as we usually do in Presburger arithmetic.

In order to recover [45, Thm. A] and more, we need to access the letters of the words in our universe.

The results of this chapter allow us to implement unary predicates L that indicate whether a word x is in the ω-regular language L (assumed to be recognised by a morphism h into an ω-semigroup M with accepting set F). The implementing automaton $\mathcal{A}$ does elementary modular arithmetic to determine, from the encodings v_n, w_n, the equivalence class $\xi \in \Xi_M$ of the n-th S-adic partial quotient $\lambda_i^{a_n} \rho_i^{b_n}$, and then simulates the automaton constructed from Thm. 4.2.4.

Proposition 4.5.1. *The first-order theory of the structure*

$$\langle \mathcal{W}; \mathsf{LowMech}, \mathsf{UpMech}, \mathsf{SameShift}, \mathsf{Char}, \mathsf{CharOrbit}, \mathsf{plus}, (L_n)_{n\in\mathbb{N}} \rangle$$

is decidable, where $\mathcal{W}$ is the set of Sturmian words over the alphabet $\{a, b\}$, and $(L_n)_{n\in\mathbb{N}}$ is the collection of ω-regular languages over $\{a, b\}$.

The above structure can express the proposition, "There exists a Sturmian word which contains a cube." It is equivalent to the formula

$$\exists x.\ \exists y_p.\ \mathsf{CharOrbit}(y_p) \wedge \neg\mathsf{Char}(y_p) \wedge \mathsf{SameShift}(y_p, x) \wedge \forall y_i \in \mathsf{CharOrbit}.\ \mathsf{Orbit}(y_i, y_p)$$
$$\Rightarrow \exists x_i, x_{i+p}, x_{i+2p}.\ \mathsf{plus}(x, y_i, x_i) \wedge \mathsf{plus}(x_i, y_p, x_{i+p}) \wedge \mathsf{plus}(x_{i+p}, y_p, x_{i+2p}) \wedge$$
$$x_i \in a\Sigma^\omega \Leftrightarrow x_{i+p} \in a\Sigma^\omega \Leftrightarrow x_{i+2p} \in a\Sigma^\omega.$$

Generalising the above constructions to Arnoux-Rauzy words, billiard words, and beyond (it seems that [16, Sec. 6] is a helpful starting point) would benefit our understanding of a broader class of shift spaces, not least because proofs of facts expressible in automatic structures can be mechanised by tools such as Walnut [57] (see also [69]) and Pecan [61]. It is thus an exciting avenue of research in logic as well as symbolic dynamics.

Chapter 5

The Monadic Theories of Arithmetic Predicates

In this chapter, we apply the results and techniques we have developed to prove the decidability of Monadic Second-Order (MSO) theories of expanded structures $\langle \mathbb{N}; <, (P_i)_{i \in I} \rangle$. The unary predicates $(P_i)_{i \in I}$ (where I is some indexing set) are of "arithmetic" nature, e.g., the powers of a for a natural number a, the set of perfect k-th powers, or the set of Fibonacci numbers. The key technical properties of these predicates is that they are ultimately periodic modulo any divisor, and can be proven "effectively sparse" by Baker's theorem.

Recall from Sec. 1.2.1 that it is equivalent to consider the characteristic word $x \in \{0, 1, \ldots, 2^d - 1\}$ of the structure, and decide whether a given (deterministic parity) automaton $\mathcal{A}$ accepts x. We outline our methods with the example of $P_0 = \{2^n \mid n \in \mathbb{N}\}$ and $P_1 = \{3^n \mid n \in \mathbb{N}\}$. For the sake of intuitive convenience, we record position 1 with the letter 1, positions of other powers of 2 with the letter 2, and positions of other powers of 3 with the letter 3, and remaining positions with the letter 0. With this encoding, the characteristic word is $x = 01232000230000002 \cdots 3 \cdots 2 \cdots$. Our plan is as follows.

1. Consider the order word $y = 12322323223 \cdots$, which is obtained by deleting intervening occurrences of the letter 0.

2. Show that for any automaton $\mathcal{A}$, we can construct a transducer $\mathcal{B}$ such that x and $\mathcal{B}(y)$ are indistinguishable to $\mathcal{A}$. This is reminiscent of the contraction method of [35]: the problem Acc_x reduces to Acc_y.

3. Identify number-theoretic dynamical systems that generate the word y. In our running example, the suffix $y(1, \infty)$ is a cutting sequence (recall Fig. 1.1,

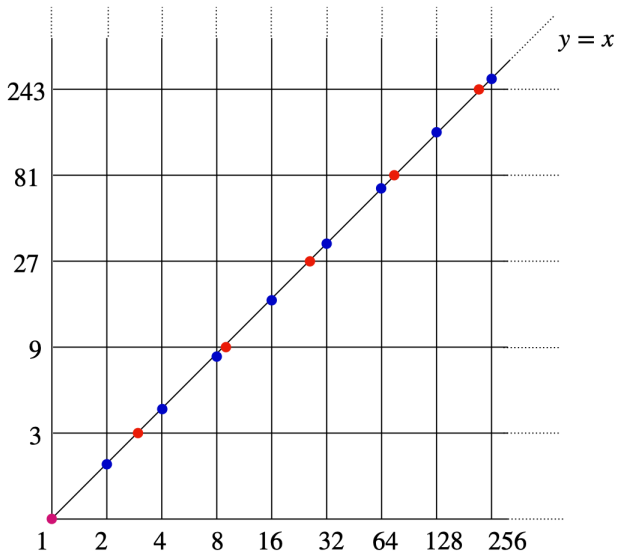

Figure 5.1: Order word as a two-dimensional cutting sequence. Cutting sequences may be generalised to higher dimensions, e.g., the word recording the order in which powers of $2, 3, 5$ occur.

reproduced for convenience) and hence an effective Sturmian word of slope $\log_6 2$ and intercept 0. More generally, higher-dimensional cutting sequences z are uniformly recurrent, but in three or more dimensions, we often need the conditional decidability of the first-order theory of $\langle \mathbb{R}; <, 0, 1, +, \cdot, \exp \rangle$ (which is subsumed by Schanuel's conjecture) in order to demonstrate the factor language of z is recursive before we can apply Cor. 2.1.4 (Semënov) to get decidability.

Step (2) can be iterated, yielding some interesting consequences. Its compositional nature gives a direct proof that the MSO theory with P_0 being the set of squares and P_1 being the set of powers of 4 is decidable; on the other hand we can prove that when P_1 is the set of powers of 2 instead, the theory reduces to the acceptance problem for the word z which gives the binary representation of $\sqrt{2}$. If we assume the conjecture that z is disjunctive, the theory is decidable by Lem. 1.2.5.

5.1 Reduction to Order Words

We describe requirements on a collection $(P_0, \ldots, P_{d-1})$ of predicates that make it amenable to Step (2) of the above outline, i.e., constructing a transducer $\mathcal{B}$ so the output $\mathcal{B}(y)$ on the order word is indistinguishable from the characteristic word x to

a given automaton $\mathcal{A}$. We use $f_i : \mathbb{N} \to \mathbb{N}$ to denote the strictly increasing function that returns the n-th element of the set P_i sorted in ascending order.

1. Each predicate P_i is *effectively procyclic*, i.e., the function f_i is effectively ultimately periodic modulo any integer divisor m.

2. The collection of predicates is *effectively sparse*, i.e., for all pairs i, j (not necessarily distinct) and k, the equation $|f_i(m) - f_j(n)| = k + 1$ has only finitely many solutions (m, n), which can moreover be effectively enumerated.

To see the significance of the first condition, consider the non-example [20, Example 5.1] of the predicate $P = \left\{ \binom{2n}{n} \mid n \in \mathbb{N}_{>0} \right\}$. By Kummer's theorem, for any prime p, the power p^k divides $\binom{2n}{n}$ if and only if there are at least k carries when performing the addition $n + n$ in base p. In particular, 4 divides $\binom{2n}{n}$ if and only if n is not a power of 2, and 3 divides $\binom{2n}{n}$ if and only if the ternary representation of n contains the digit 2. This allows us to encode an open conjecture of Erdős which states that for every $n > 8$, the ternary representation of 2^n contains the digit 2. See also [41, Prob. B33] for related open problems we can encode with this predicate.

The second condition is crucial to the indistinguishability argument. It implies that if the characteristic word x is factorised as $0^{k_0}y(0)0^{k_1}y(1)\cdots$, where y is the order word, then for any threshold k, we can compute N such that for all $n \geq N$, the number k_n of intervening 0s before the letter $y(n)$ in x exceeds k.

The divisor m and threshold k are derived from the language L recognised by the given automaton $\mathcal{A}$. More precisely, let L be recognised by a morphism h into an ω-semigroup M with accepting set F. By a standard counting argument, we have that there exist m, k such that for all $n \geq k$, it holds that $h(0^n) = h(0^{n+m})$. We assume without losing generality that m divides k.

The transducer $\mathcal{B}$ uses its finite memory and the effectively procyclic property to determine the remainder r_i when k_i is divided by m, and thus insert $k + r_i$ intervening 0s in the suffix where k_i is guaranteed to be large enough. Formally, we have the following.

Lemma 5.1.1. *Let $(P_0, \ldots, P_{d-1})$ be an effectively sparse collection of effectively procyclic predicates; let x, y respectively be the attendant characteristic and order words. Given an automaton $\mathcal{A}$, we can construct a transducer $\mathcal{B}$ such that $\mathcal{A}$ accepts x if and only if it accepts $\mathcal{B}(y)$. Thus, Acc_x and Acc_y are Turing-equivalent.*

Proof. Let the divisor m and threshold $k = dm$ computed from the automaton $\mathcal{A}$ be as above. We also compute N such that for all $n \geq N$, the number k_n of intervening 0s before the letter $y(n)$ in x exceeds k.

States of the transducer $\mathcal{B}$ have the following components:

1. $\mathsf{lastRemainder}$, which gives the remainder when n, the index in x of the last read letter from y, is divided by m (initialised to $m-1$);

2. $\mathsf{numIntervals}$, which is a value in $\{0, 1, \ldots, N-1, \mathsf{enough}\}$, to track how many letters of y we have read (initialised to 0);

3. phase_i for each i, which captures the procyclic behaviour of f_i by indicating the remainder when n, the next index at which P_i holds, is divided by m (each initialised to $f_i(0) \bmod m$).

Let us describe the transitions taken and words output upon reading a letter $c = \sum_{i=0}^{d-1} b_i 2^i$, where at least one b_i is 1.

1. The $\mathsf{numIntervals}$ component gets incremented from j to $j+1$ for $j < N-1$, and from $N-1$ to enough, and stays constant at enough.

2. The $\mathsf{lastRemainder}$ component is updated to $\mathsf{currentRemainder} = \mathsf{phase}_i$, which will be the same for all i such that $b_i = 1$.

3. For each i such that $i = 1$, the component phase_i advances to the next phase. (If one is to be technically precise, for each i, we have a divisor d and a threshold l such that the sequence of phases is the ultimately periodic sequence $f_i(0), f_i(1), \ldots, f_i(l), f_i(l+1), , \ldots, f_i(l+d-1), f_i(l), f_i(l+1), \ldots$ modulo m.)

The word printed upon reading $y(n)$ is $0^{k_n} y(n)$ if $n < N$ (i.e., $\mathsf{numIntervals}$ is not enough); otherwise it is $0^{k+r} y(n)$, where

$$r \equiv \mathsf{currentRemainder} - \mathsf{lastRemainder} - 1 (\bmod\ m).$$

It follows by construction that x and $\mathcal{B}(y)$ are indistinguishable to $\mathcal{A}$. By Lem. 1.2.2, we then have that Acc_x reduces to Acc_y. Of course, Acc_y is a special case of Acc_x, and the stated Turing equivalence follows. □

We record an interesting composability lemma which follows as a corollary of the above proof. Taking P_1 to be the set of squares, and P_2 to be the set of powers of 2 would give the decidability of the MSO theory of $\langle \mathbb{N}; <, \{n^2 \mid n \in \mathbb{N}\}, \{4^n \mid n \in \mathbb{N}\} \rangle$.

Lemma 5.1.2. *Let $P_1, \ldots, P_d$ be effectively procyclic predicates such that each collection among $(P_1), \ldots, (P_d)$ is effectively sparse. Let $g_1, \ldots, g_d$ denote functions such that g_i returns the n-th element of P_i. For $i \leq j$, let $f_{i,j} = g_i \circ \cdots \circ g_j$. Define $R_{i,j} = \{f_{i,j}(n) \mid n \in \mathbb{N}\}$. The MSO theory of $\langle \mathbb{N}; <, R_{1,1}, \ldots, R_{1,d} \rangle$ is decidable.*

Proof. Let x_i, y_i respectively denote the characteristic and order words of $\langle \mathbb{N}; <, R_{i,i}, \ldots, R_{i,d} \rangle$. One checks by definition that $y_i = x_{i+1}$ for $1 \leq i < d$ (modulo renaming the letters). By Lem. 5.1.1, we thus have that Acc_{x_i} reduces to $\mathsf{Acc}_{x_{i+1}}$ for $1 \leq i < d$, and Acc_{x_d} is decidable by virtue of reducing to the straightforward acceptance problem for the order word 1^ω. Assembling the chain of reductions proves that the MSO theory of $\langle \mathbb{N}; <, R_{1,1}, \ldots, R_{1,d} \rangle$ is decidable. □

We now give examples of effectively sparse collections of effectively procyclic predicates. The first is due to a classical result by Schinzel and Tijdeman [67].

Theorem 5.1.3 (Schinzel-Tijdeman)**.** *Let b, d, p, q be positive integers with $b, d \geq 2$. For every $k \geq 1$, the equation $|qn^d - pb^m| = k$ has finitely many solutions $(n, m) \in \mathbb{N}^2$ which can moreover be effectively enumerated. Thus, the collection*

$$(\{qn^d \mid n \in \mathbb{N}\}, \{pb^n \mid n \in \mathbb{N}\})$$

of effectively procyclic predicates is effectively sparse.

Theorems such as the above are typically proven by using the theory of linear forms in logarithms, whose development earned Alan Baker the Fields Medal in 1970. Several progressively sharper formulations of Baker's theorem have been established since; in this thesis, we will rely on (a special case of) the expository version [7, Thm. 3.1].

Recall that a number is said to be *algebraic* if it is a root of a polynomial with integer coefficients; every algebraic number α is the root of a unique irreducible polynomial p (i.e., the polynomial does not factorise over the integers), the gcd of whose coefficients is 1. We say that p is the *minimal polynomial* of α, its degree d is the *degree* of α, and the maximum H of the absolute values of the coefficients of p is called the *height* of α. (More sophisticated notions of height exist, but the one we introduce suffices for our discussion.)

Theorem 5.1.4 (Baker)**.** *Let $\alpha_1, \ldots, \alpha_n$ be real positive algebraic numbers with degrees at most d and heights at most A. Let $b_1, \ldots, b_n$ be integers of modulus at most $B \geq 2$. Consider the quantity*

$$\Lambda = b_1 \log \alpha_1 + \cdots + b_n \log \alpha_n.$$

If $\Lambda \neq 0$, then $|\Lambda| > B^{-C}$, where C is an effectively computable number depending only on n, d, A.

We apply the theorem to prove the following effective sparsity lemma.

Lemma 5.1.5. *Let $\alpha_1, \alpha_2, c_1, c_2$ be real positive algebraic numbers with $\alpha_1, \alpha_2 \geq 1$. For $i \in \{1, 2\}$, define $f_i : \mathbb{N} \to \mathbb{R}$ as $f_i(n) = c_i \alpha_i^n$, and define $g_i : \mathbb{N} \to \mathbb{R}$ as*

$$g_i(n) = \min_{\{m | f_{3-i}(m) > f_i(n)\}} f_{3-i}(m) - f_i(n),$$

i.e., $g_i(n)$ gives the gap to the smallest value of f_{3-i} larger than $f_i(n)$. Dually, define $h_i(n)$ as the gap to the largest value of f_{3-i} smaller than $f_i(n)$. We can compute polynomials P_i, Q_i such that $g_i(n) > f_i(n)/P_i(n)$ and $h_i(n) > f_i(n)/Q_i(n)$.

Proof. We show how to lower bound $\Theta = c_1\alpha_1^{n_1} - c_2\alpha_2^{n_2}$, given that it is positive. Here, n_1 is chosen minimally given n_2, and n_2 is chosen maximally given n_1. These choices are respectively upper bounded by

$$n_1 \leq L_2(n_2) = \lceil \log \alpha_2 / \log \alpha_1 \rceil \cdot n_2 + \lceil \log(c_2/c_1) / \log \alpha_1 \rceil + 1 \geq n_2$$

and

$$n_2 \leq L_1(n_1) = \lceil \log \alpha_1 / \log \alpha_2 \rceil \cdot n_1 + \lceil \log(c_1/c_2) / \log \alpha_2 \rceil + 1 \geq n_1.$$

We now rewrite

$$\Theta = f_2(n_2) \left(\frac{c_1}{c_2} \cdot \frac{\alpha_1^{n_1}}{\alpha_2^{n_2}} - 1 \right) = f_1(n_1) \left(1 - \frac{c_2}{c_1} \cdot \frac{\alpha_2^{n_2}}{\alpha_1^{n_1}} \right).$$

We use the facts that $\exp(x) - 1 > x > 0$ and $1 - \exp(-x) > \frac{x}{1+x} > 0$ for positive x. Let $\Lambda > 0$ denote $(1 \cdot \log(c_1/c_2) + n_1 \cdot \log \alpha_1 - n_2 \cdot \log \alpha_2)$. We thus have

$$g_2(n_2) > f_2(n_2) \cdot \Lambda$$

and

$$h_1(n_1) > f_1(n_1) \cdot \frac{\Lambda}{1 + \Lambda}.$$

Applying Baker's theorem gives $\Lambda > 1/(L_1(n_1))^C$ and $\Lambda > 1/(L_2(n_2))^C$. Since both $x \mapsto x$ and $x \mapsto \frac{x}{1+x}$ are increasing in x for positive real input, we can substitute the lower bounds on Λ to obtain

$$g_2(n_2) > \frac{f_2(n_2)}{(L_2(n_2))^C}$$

and

$$h_1(n_1) > \frac{f_1(n_1)}{1 + (L_1(n_1))^C},$$

as desired. □

Proposition 5.1.6. *Any collection* $\{a_1^n \mid n \in \mathbb{N}\}, \ldots, \{a_d^n \mid n \in \mathbb{N}\}$ *given by integers* $a_1, \ldots, a_d$ *is effectively sparse.*

Proof. One simply applies the preceding lemma to each pair. □

We may also consider other effectively procyclic predicates with exponential growth rates, e.g., the Fibonacci sequence: indeed, a natural class comprises *linear recurrence sequences (LRS)* with a single real positive *dominant root* which is moreover *simple*.

Let us recall the relevant definitions. An (integer) LRS $u : \mathbb{N} \to \mathbb{Z}$ of order d is given by its initial terms $u(0), \ldots, u(d-1)$, and the integer recurrence relation

$$u(n+d) = a_{d-1}u(n+d-1) + \cdots + a_0 u(n)$$

that it satisfies. (I.e., the coefficients $a_{d-1}, \ldots, a_0$ are integers with $a_0 \neq 0$.) Its characteristic polynomial is $X^d - a_{d-1}X^{d-1} - \cdots - a_0$; the algebraic roots of this polynomial are called characteristic roots, and those with the maximum modulus are called dominant roots. A root λ_i is simple if its multiplicity m_i is 1.

An LRS has the closed form expression

$$u(n) = \sum_i \sum_{j=0}^{m_i - 1} p_{ij} n^j \lambda_i^n.$$

The following is proven by repeatedly applying Lem. 5.1.5 to argue that beyond a certain index, the contribution of the non-dominant terms is too negligible to bridge the gap between the dominant terms from the sequences, and hence the non-dominant terms eventually stop playing a role in determining the order word.

Proposition 5.1.7. *Let* $u_1, \ldots, u_d : \mathbb{N} \to \mathbb{N}$ *be LRS, each with a single simple real positive dominant root* ρ_i*, and let* $u_i(n) = c_i\rho_i^n + r_i(n)$*. Assume that we do not have any pair* ρ_i, ρ_j *such that* $\rho_i^m = \rho_j^n$ *for some* m, n*. For every* k*, we can compute* N *such that at arbitrary indices* $n_1, \ldots, n_d$ *larger than* N*:*

1. $|u_i(n_i) - u_j(n_j)| > k$*;*
2. $u_i(n_i) > u_j(n_j)$ *if and only if* $c_i\rho_i^{n_i} > c_j\rho_j^{n_j}$*.*

Let P_i *denote the predicate* $\{u_i(n) \mid n \in \mathbb{N}\}$*. The MSO theory of* $\langle \mathbb{N}; <, P_1, \ldots, P_d \rangle$ *thus reduces to the acceptance problem for the attendant order word* y*. In turn, beyond an effectively computable index, the word* y *records the relative order of the algebraic numbers* $c_i\rho_i^{n_i}$*.*

Let us remark that the restriction on the dominant roots to be pairwise multiplicatively independent is technically necessary. Indeed, if we were allowed to have predicates corresponding to both $\rho^n + u(n)$ and ρ^n, where ρ^n grows much faster than the LRS $u(n)$, then the attendant order word would be of the form $(ab)^\omega$ if and only if $u(n)$ is positive for all n, and the order word would be of the form $(\Sigma\Sigma)^*(ab)^\omega$ if and only if $u(n)$ is positive for all but finitely many n. These queries respectively correspond to the Positivity and Ultimate Positivity problems for LRS, which are known to reduce to open problems in Diophantine approximation [62]. Barring number-theoretic breakthroughs, this rules out an algorithm that runs uniformly on the class of unrestricted theories; in fact there are specific instances of LRS whose Positivity is unknown.[1]

5.2 Automata on Almost-Periodic Order Words

In this section, we will study order words of structures $\langle \mathbb{N}; <, P_0, \ldots, P_{d-1} \rangle$ where the unary predicates $P_0, \ldots, P_{d-1}$ are sets of values of LRS with a single positive (simple) dominant root (as in Prop. 5.1.7). We have already seen, e.g., that the order word which records the order in which the powers of 2 and 3 occur is a Sturmian word of slope $\log_6 2$, and is hence effectively uniformly recurrent (Lem. 1.1.13). Consequently, this word has a decidable automaton acceptance problem (Cor. 2.1.4). We thus have the following result from the abstract.

Proposition 5.2.1. *Let $\rho_1, \rho_2 \in \mathbb{N}$. The MSO theory of the structure $\langle \mathbb{N}; <, \{\rho_1^n \mid n \in \mathbb{N}\}, \{\rho_2^n \mid n \in \mathbb{N}\} \rangle$ is decidable.*

When ρ_1, ρ_2 are multiplicatively independent, the order word is a Sturmian word of slope $\log_{\rho_1\rho_2} \rho_1$. Otherwise, this quantity is rational, and there exists $\rho \in \mathbb{N}$ such that both ρ_1, ρ_2 are powers of ρ. In this case, the characteristic word of the structure is obtained by a simple transduction on the characteristic word of $\langle \mathbb{N}; <, \{\rho^n \mid n \in \mathbb{N}\} \rangle$. Decidability then follows by classic arguments [20] and Lem. 1.2.2.

We will generalise the above as follows.

Theorem 5.2.2. *Let Pow_ρ denote the predicate $\{\rho^n \mid n \in \mathbb{N}\}$. The problem of deciding the MSO theory of the structure $\langle \mathbb{N}; <, (\mathsf{Pow}_\rho)_{\rho \in \mathbb{N}} \rangle$ reduces to deciding the first-order theory of the structure $\langle \mathbb{R}; <, +, \cdot, 0, 1, \exp \rangle$.*

We also have a related result which is proven using the same ideas.

[1] working notes of the Foundations of Algorithmic Verification group led by Joël Ouaknine.

Theorem 5.2.3. *Let $u_1, \ldots, u_d : \mathbb{N} \to \mathbb{N}$ be LRS, each with a single simple real positive dominant root ρ_i, and let $u_i(n) = c_i\rho_i^n + r_i(n)$. Assume that we do not have any pair ρ_i, ρ_j such that $\rho_i^m = \rho_j^n$ for some m, n. Let P_i denote the predicate $\{u_i(n) \mid n \in \mathbb{N}\}$. The problem of deciding the MSO theory of the structure $\langle \mathbb{N}; <, P_1, \ldots, P_d\rangle$ reduces to deciding the first-order theory of the structure $\langle \mathbb{R}; <, +, \cdot, 0, 1, \exp\rangle$.*

Observe that in the case of Thm. 5.2.2, the concrete input to the algorithm is a sentence that can be interpreted in some finite-signature structure $\langle \mathbb{N}; <, P_1, \ldots, P_d\rangle$. We can furthermore assume, by the transduction argument of the preceding paragraph, that these predicates satisfy the premises of Prop. 5.1.7. The decision problems thus reduce to the automaton-acceptance problems for the corresponding order words.

We will observe that these order words (modulo an effectively computable prefix) are cutting sequences, argue that cutting sequences are uniformly recurrent, and show that they are effectively uniformly recurrent if we assume that the first-order theory of the structure $\langle \mathbb{R}; <, +, \cdot, 0, 1, \exp\rangle$ is decidable.

5.2.1 Uniform Recurrence of Cutting Sequences

The word recording the order of elements of the sets $\{c_1\rho_1^n \mid n \in \mathbb{N}\}, \ldots, \{c_d\rho_d^n \mid n \in \mathbb{N}\}$ can be realised a a cutting sequence as follows. Consider the Euclidean space $\mathbb{R}^d_{\geq 0}$, the ray aligned with the vector $(1, \ldots, 1)$, and the map $(x_1, \ldots, x_d) \mapsto (\exp x_1, \ldots, \exp x_d)$. We construct the hyperplanes $x_i = \log c_i + n \log \rho_i$ for each *orientation* $i \in \{1, \ldots, d\}$ and iterate $n \in \mathbb{N}$. Clearly, the cutting sequence $y \in \{1, \ldots, d\}^\omega$ which records the orientations of the sequence of hyperplanes the ray intersects, starting from the origin, is the order word. In the sequel, it will be helpful to intuit $(1, \ldots, 1)$ as a velocity vector.

Cutting sequences are readily seen to have uniformly recurrent suffixes by virtue of their toric nature. Let us rescale our space so that parallel hyperplanes are unit distance apart. Our cutting ray is then aligned with the vector $(1/\log \rho_1, \ldots, 1/\log \rho_d)$. (Indeed, the "time" taken to traverse distance $\log \rho$ at "velocity" 1 in the original coordinates should be the same as that taken to traverse distance 1 in the new coordinates.) We distinguish the orientation $i = 1$; we may assume without losing generality that $\rho_1 \leq \rho_i$ for all i. We shall factorise the cutting sequence[2] $y \in \{1, \ldots, d\}^\omega$ as $r_0 r_1 \cdots$ based on return words to the letter 1, and observe that y is the image of a toric word z under a substitution τ with $r_n = \tau(z(n))$ for each n.

[2] technically, its suffix

We consider the suffix of y which records intersections from a point where hyperplanes of all orientations have been crossed earlier, and the cutting line is guaranteed to never cross two hyperplanes simultaneously. The arrangement of these hyperplanes partitions the space into "unit cells" that may each be identified with $[0,1)^d$; likewise they partition each hyperplane into unit cells that we can identify with $[0,1)^{d-1}$. Our toric word will be based on the (relative) coordinates of the unit cell where hyperplanes of orientation $i = 1$ are intersected by the cutting ray.

In more technical terms, the unit cell $[0,1)^{d-1}$ is mapped to the torus $(S^1)^{d-1}$, and the discrete translation is $(\log \rho_1/\log \rho_2, \ldots, \log \rho_1/\log \rho_d)$ — this is (the projection of) the vector traversed by the cutting ray between intersections with hyperplanes of orientation $i = 1$.

Clearly, any letter i in our suffix of the cutting sequence has only finitely many return words, because the ray can intersect hyperplanes of other orientations only finitely many times between consecutive intersections with the same orientation. Thus, when we partition our torus based on return words, there will be only finitely many partitions. In our specific case of $i = 1$, return words may have at most one occurrence of each letter.

These partitions are determined by linear programming. For instance, the letter j will feature in the current factor of the return word only if the current point $(\beta_2, \ldots, \beta_d)$ of the orbit is such that in the next step, the increment of $\log \rho_1/\log \rho_j$ causes the j-th coordinate to "wrap around." Points such that $\beta_j > 1 - \log \rho_1/\log \rho_j$ will fall into partitions that correspond to return words that contain the letter j.

Linear programming also determines the order in which multiple letters occur in a return word. Intuitively, if both the i-th and the j-th coordinates "wrap around", we need to identify which happened "first" in the cutting sequence picture. The kinematics-based intuition is helpful again: in the i-th coordinate, the distance is $1 - \beta_i$, and the velocity is $1/\log \rho_i$. If the letter j is to occur first, we must have that $(1 - \beta_i) \log \rho_i > (1 - \beta_j) \log \rho_j$. The reader may note that this generalises the observation of the preceding paragraph, whose condition may be rearranged as $(1) \log \rho_1 > (1 - \beta_j) \log \rho_j$.

We illustrate the above arguments with the example of the powers of $2, 3, 5$ in Fig. 5.2.

The uniform recurrence of cutting sequences follows from standard compactness arguments. Consider a translation $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$ on the torus $(S^1)^d$. The (sub)torus $\mathbb{T}$ *generated* by $\boldsymbol{\alpha}$ is the topological closure of the set $\{n\boldsymbol{\alpha} \mid n \in \mathbb{N}\}$. This space has

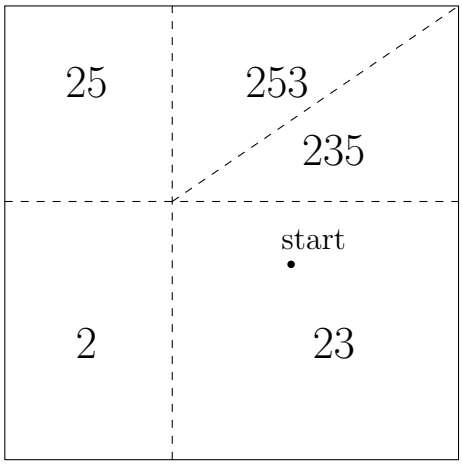

Figure 5.2: Return-word factorisation of the order word marking powers of $2, 3, 5$. For clarity, we use the base ρ_i itself to denote its corresponding letter. The boundaries of the partitions intersect at $\left(1 - \frac{\log 2}{\log 3}, 1 - \frac{\log 2}{\log 5}\right)$. The starting point is $\left(\frac{\log 2}{\log 3}, \frac{\log 2}{\log 5}\right)$.

an elegant characterisation in terms of the $\mathbb{Z}$-module

$$\mathbb{L} = \{\mathbf{l} = (l_1, \ldots, l_d) \in \mathbb{Z}^d \mid \mathbf{l} \cdot \boldsymbol{\alpha} = \sum_i l_i \alpha_i \in \mathbb{Z}\}$$

of linear dependencies of $\boldsymbol{\alpha}$, which has a finite basis. The following is folklore (see, e.g., [39]).

Proposition 5.2.4 (Kronecker)**.** *We have that*

$$\mathbb{T} = \{\boldsymbol{\beta} \in (S^1)^d \mid \mathbf{l} \cdot \boldsymbol{\beta} \in \mathbb{Z} \text{ for all } \mathbf{l} \in \mathbb{L}\}.$$

Furthermore, for every $\boldsymbol{\beta} \in \mathbb{T}$ and $\varepsilon > 0$, there exist infinitely many $n \in \mathbb{N}$ such that $n\boldsymbol{\alpha}$ is ε-close to $\boldsymbol{\beta}$.

Thus, given a starting point $\boldsymbol{\gamma}$ on the torus, the infinite orbit obtained by translating by $\boldsymbol{\alpha}$ is dense in the compact set $\boldsymbol{\gamma} + \mathbb{T}$. The standard argument to establish the uniform recurrence of toric words then proceeds as follows.

For each factor u, we define an open region I_u: for the base cases, this region corresponds to the partition of the ambient torus $(S^1)^d$, the inductive step is to define I_{vu} as $I_v \cap (-|v| \cdot \boldsymbol{\alpha} + I_u)$. By definition, there is an occurrence of factor u at index n if and only if the orbit is in I_u at the n-th step. We will argue that if the orbit does visit I_u, then there necessarily exists a return time R_u such that consecutive visits to I_u are at most I_u steps apart.

Indeed, by Prop. 5.2.4[3], the family $(I_u - n\boldsymbol{\alpha})_{n \in \mathbb{N}}$, whose n-th member is the set of points that reach I_u in n steps, is an open cover of the compact orbit-closure $\boldsymbol{\gamma} + \mathbb{T}$.

[3]Here and elsewhere, we implicitly use the abelian group structure of the torus.

By the definition of compactness, this open cover necessarily admits a finite subcover: number R_u of its members is the desired return time.

5.2.2 Effective Uniform Recurrence of Cutting Sequences

We now need to show how the particular toric words we consider are *effectively* uniformly recurrent. Recall that this is equivalent to deciding, given a word u, whether u occurs as a factor. A semi-algorithm to answer YES simply enumerates increasingly long prefixes. It is trickier to design the dual semi-algorithm that answers NO: our strategy is to establish that $I_u \cap (\boldsymbol{\gamma} + \mathbb{T})$ is empty.

Lemma 5.2.5. *We can decide whether I_u is empty.*

Proof. There are finitely many minimal ways to extend u into a concatenation $r_0 r_1 \cdots r_k$ of return words that indicate which partitions of the torus the trajectory must fall into at the current step, and in the k subsequent steps. Let $I_0, I_1, \ldots, I_k$ denote these polytopes. Let the coordinates of the current step be $(\beta_2, \ldots, \beta_d)$, and recall the translation is $\left(\frac{\log \rho_1}{\log \rho_2}, \ldots, \frac{\log \rho_1}{\log \rho_d}\right)$. Let us introduce binary-valued variables $b_{i,j}$ for $i \in \{2, \ldots, d\}$ and $j \in \{1, \ldots, k\}$ to indicate whether there was a "wrap-around" in the i-th coordinate while performing the translation from the $(j-1)$-st step to the j-th step. Formally, the i-th coordinate of the point which encodes r_j is $\beta_i + j \cdot \frac{\log \rho_1}{\log \rho_i} - \sum_{a=1}^{j} b_{i,a}$. We will test the emptiness of I_u by checking the feasibility of finitely many linear programs, each corresponding to some choice of concatenation $r_0 \cdots r_k$, and binary variables $b_{i,j}$. We use $m_{i,j}$ to denote the quantity $\sum_{a=1}^{j} b_{i,a}$, which is constant in each linear program. (Note that $m_{i,0} = 0$.)

The linear program checks whether for each $j \in \{0, 1, \ldots, k\}$, we have

$$\left(\ldots, \beta_i + j \cdot \frac{\log \rho_1}{\log \rho_i} - m_{i,j}, \ldots \right) \in I_j.$$

This yields inequalities of the form

$$0 < \beta_i \log \rho_i + j \cdot \log \rho_1 - m_{i,j} \log \rho_i < \log \rho_i$$

from the bounds on coordinates, and

$$\log \rho_i - \beta_i \log \rho_i + m_{i,j} \log \rho_i > \log \rho_a - \beta_a \log \rho_a + m_{a,j} \log \rho_a$$

and

$$\log \rho_1 > \log \rho_i - \beta_i \log \rho_i - j \log \rho_1 + m_{i,j} \log \rho_j$$

from partitioning-polytope constraints. Observe that the coefficient of β_i is always an integer multiple of $\log \rho_i$: we can replace $\beta_i \log \rho_i$ with a fresh variable β'_i. Observe also that the constant terms are linear forms in $\log \rho_1, \ldots, \log \rho_d$. Thus, performing Fourier-Motzkin elimination reduces checking the feasibility of such a conjunction of constraints to checking whether a conjunction of inequalities in linear forms of logarithms holds, i.e., inequalities of the form $a_1 \log \rho_1 + \cdots a_d \log \rho_d > 0$.

One may elicit the linear dependencies between $\log \rho_1, \ldots, \log \rho_d$ — these correspond to multiplicative dependencies between the algebraic numbers $\rho_1, \ldots, \rho_d$, which can be found using [23] — and thus check whether the linear form is 0. If it is nonzero, then Baker's theorem (Thm. 5.1.4) prescribes how precisely one must numerically approximate this form to determine its sign. □

Deciding the emptiness of I_u suffices when $\mathbb{T} = (S^1)^{d-1}$, or in other words, the numbers $1, \frac{\log \rho_1}{\log \rho_2}, \ldots, \frac{\log \rho_1}{\log \rho_d}$ are $\mathbb{Z}$-linearly independent. Such is provably the case when the vector space of rational linear combinations of $\log \rho_1, \ldots, \log \rho_d$ has dimension 2. As an example, consider $(\rho_1, \rho_2, \rho_3) = (2, 3, 6)$.

Lemma 5.2.6. *Let $\rho_1, \ldots, \rho_d$ be such that there exist multiplicatively independent real positive algebraic μ_1, μ_2 such that for every $i \in \{1, \ldots, d\}$, there exist $a_i, b_i \in \mathbb{Z}$ such that $\log \rho_i = a_i \log \mu_1 + b_i \log \mu_2$. We have that the numbers $\frac{1}{\log \rho_1}, \ldots, \frac{1}{\log \rho_d}$ are $\mathbb{Z}$-linearly independent.*

Proof. Suppose, for the sake of deriving a contradiction, that there exist integers $c_1, \ldots, c_d$, not all zero, such that

$$\frac{c_1}{\log \rho_1} + \cdots + \frac{c_d}{\log \rho_d} = 0.$$

We can multiply throughout by $\log \rho_1 \cdots \log \rho_d$, and substitute each $\log \rho_i$ with $a_i \log \mu_1 + b_i \log \mu_2$ to get $P(\log \mu_1, \log \mu_2) = 0$, where $P \in \mathbb{Z}[X_1, X_2]$ is a nonzero homogenous polynomial of degree $d-1$. The number $\zeta = \frac{\log \mu_1}{\log \mu_2}$ is thus algebraic, which contradicts the following result of Baker [70, Thm. 1.6].

Theorem 5.2.7. *Let $\rho_1, \ldots, \rho_d$ be multiplicatively independent real positive algebraic numbers. We then have that $1, \log \rho_1, \ldots, \log \rho_d$ are linearly independent over the field of algebraic numbers.*

In our case, it implies that we cannot have $\log \mu_1 - \zeta \log \mu_2 = 0$, completing the proof by contradiction. □

Consequently, we have the following special case of Thm. 5.2.3.

Theorem 5.2.8. *Let $u_1, \ldots, u_d : \mathbb{N} \to \mathbb{N}$ be LRS, each with a single simple real positive dominant root ρ_i, and let $u_i(n) = c_i\rho_i^n + r_i(n)$. Assume that we do not have any pair ρ_i, ρ_j such that $\rho_i^m = \rho_j^n$ for some m, n. Assume also that the vector space of rational linear combinations of $\log \rho_1, \ldots, \log \rho_d$ has dimension 2. Let P_i denote the predicate $\{u_i(n) \mid n \in \mathbb{N}\}$. The MSO theory of the structure $\langle \mathbb{N}; <, P_1, \ldots, P_d\rangle$ is decidable.*

In the more general case, e.g., $(\rho_1, \rho_2, \rho_3) = (2, 3, 5)$, one's intuition would continue to expect that $\frac{1}{\log 2}, \frac{1}{\log 3}, \frac{1}{\log 5}$ are linearly independent over the rationals, because "there is no conceivable reason for them not to be." Perhaps surprisingly, this is yet to be proven — it is an instance of Schanuel's conjecture [50, Pages 30-31].

Conjecture 5.2.9 (Schanuel)**.** *If the complex numbers $\zeta_1, \ldots, \zeta_d$ are linearly independent over $\mathbb{Q}$, then the set $\{\zeta_1, \ldots, \zeta_d, \exp \zeta_1, \ldots, \exp \zeta_d\}$ contains a subset of d algebraically independent numbers.*

Recall that numbers $\gamma_1, \ldots, \gamma_d$ are algebraically independent if there does not exist any polynomial $P \in \mathbb{Z}[X_1, \ldots, X_d]$ such that $P(\gamma_1, \ldots, \gamma_d) = 0$. For example, the open question of the transcendence of $e+\pi$ and $e\pi$ would immediately be answered in the affirmative if it were proven that Euler's constant e and the circle constant π are algebraically independent. Schanuel's conjecture readily implies this independence, as $\exp 1 = e, \exp(i\pi) = -1$, and the only algebraically independent pair among $1, i\pi, e, -1$ can be $i\pi, e$.

In our instance, the numbers $\log 2, \log 3, \log 5$ are clearly linearly independent over the rationals: if they were not, we would have the absurd existence of integers a, b, c, not all zero, such that $2^a 3^b 5^c = 1$. If there is a triple of algebraically independent numbers among $\log 2, \log 3, \log 5, 2, 3, 5$, it can only be $\log 2, \log 3, \log 5$, and Schanuel's conjecture asserts that it indeed is. In particular, this implies the linear independence of $\frac{1}{\log 2}, \frac{1}{\log 3}, \frac{1}{\log 5}$ over the rationals. Consequently for the attendant order word, the torus $\mathbb{T} = (S^1)^2$, and deciding the emptiness of I_u suffices for effective uniform recurrence if we assume Schanuel's conjecture.

Remark 5.2.10. *We have thus proven that the MSO theory of the structure*

$$\langle \mathbb{N}; <, \{2^n \mid n \in \mathbb{N}\}, \{3^n \mid n \in \mathbb{N}\}, \{5^n \mid n \in \mathbb{N}\}\rangle$$

is decidable subject to Schanuel's conjecture, as stated in the abstract.

In the rest of the section, we will refine this intermediate result to the statements of Thms. 5.2.2 and 5.2.3. The first-order theory of the structure $\langle \mathbb{R}; <, +, \cdot, 0, 1, \exp \rangle$ mentioned in these statements was shown by Macintyre and Wilkie [53] to be decidable subject to Schanuel's conjecture. (We shall henceforth denote this theory by $\mathcal{T}_{\exp}$ for brevity; we likewise denote the logical signature by $\mathcal{S}_{\exp}$.) Their algorithm — unlike the one we provide for our intermediate result above — is unconditionally correct, and requires Schanuel's conjecture only for proof of termination. In fact, Macintyre and Wilkie further proved that the decidability of $\mathcal{T}_{\exp}$ is *equivalent* to a hypothesis that is similar in spirit to, but ostensibly weaker than, Schanuel's conjecture.

Recall that we have already reduced the decidability of the MSO theory to the effective uniform recurrence of (some computable suffix of) the order word, and we may readily certify that a factor u appears by simply enumerating prefixes of the order word. For the dual semi-algorithm, we now need to show how we semi-decide, via queries to $\mathcal{T}_{\exp}$, whether $I_u \cap (\boldsymbol{\gamma} + \mathbb{T})$ is empty.

In other words, the following lemma would complete the proofs of Thms. 5.2.2 and 5.2.3.

Lemma 5.2.11. *The emptiness of $I_u \cap (\boldsymbol{\gamma} + \mathbb{T})$ is entailed by a sentence in $\mathcal{T}_{\exp}$ with the semantics*

$$\neg\exists \boldsymbol{\beta}.\ \left((\boldsymbol{\beta} \in I_u) \wedge \bigwedge_i (\mathbf{l}_i \cdot (\boldsymbol{\beta} - \boldsymbol{\gamma}) \in \mathbb{Z}) \right),$$

where the vectors $\mathbf{l}_i$ of integer constants are elements of the basis of the $\mathbb{Z}$-module $\mathbb{L}$ (it is not necessary all elements feature in the conjunction), and are enumerated by querying $\mathcal{T}_{\exp}$ with variable-free sentences of the form

$$\mathbf{l} \cdot \boldsymbol{\alpha} = c.$$

Proof. The strategy to construct a sentence with the required entailment property, and the entailment of emptiness itself, are immediate. The outstanding concern is the expressibility of these sentences in $\mathcal{S}_{\exp}$.

More specifically, we need to demonstrate that the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ of constant transcendental entries can be defined in $\mathcal{S}_{\exp}$. Note that while constructing formulas, atomic propositions of the form $\mathsf{term}(\boldsymbol{\beta}) \in \mathbb{Z}$ can be decomposed into finite disjunctions because the variables are bounded, and the coefficients are known constants.

To define $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, note that their transcendental entries are (ratios of) logarithms of real positive algebraic numbers. It is well known that algebraic numbers can be defined in the first-order signature of the structure $\langle \mathbb{R}; <, +, \cdot, 0, 1 \rangle$, e.g., as the root of its minimal polynomial distinguished by a rational approximation and a radius of

precision. Having forced a variable ζ to hold an algebraic number ζ_0, we force the variable ξ to hold the constant $\log \zeta_0$ using the atomic proposition $\zeta = \exp \xi$. □

5.3 Automata on High-Entropy Order Words

In this section, we take the reduction given by Thm. 5.1.3 as our starting point. We study the decidability of MSO theories of structures of the form

$$\langle \mathbb{N}; <, \{qn^d \mid n \in \mathbb{N}\}, \{pb^n \mid n \in \mathbb{N}\} \rangle$$

via the acceptance problems corresponding to their respective order words.

Proposition 5.3.1. *Let $x \in (\{0,1\}^2)^\omega$ be the order word of the structure*

$$\langle \mathbb{N}; <, \{qn^d \mid n \in \mathbb{N}\}, \{pb^n \mid n \in \mathbb{N}\} \rangle,$$

and let $y \in (\{0, 1, \ldots, b-1\}^d)^\omega$ be the word whose components indicate the digits of the base-b expansions of the numbers $\sqrt[d]{p/q}, \sqrt[d]{pb/q}, \ldots, \sqrt[d]{pb^{d-1}/q}$. We have that Acc_x and Acc_y are Turing-equivalent.

In the above, the word y is synchronised, i.e., there is an offset l such that for each r, we have $\sqrt[d]{pb^r/q} = \sum_{i=0}^{\infty} y(i)(r) \cdot b^{l-i}$.

The following corollary, which was stated in the abstract, inspires the title of this section and is an immediate consequence of the above proposition and Lem. 1.2.5. (Note that if b is a perfect square, decidability follows from Lem. 5.1.2.)

Corollary 5.3.2. *The MSO theory of the structure*

$$\langle \mathbb{N}; <, \{n^2 \mid n \in \mathbb{N}\}, \{b^n \mid n \in \mathbb{N}\} \rangle$$

is decidable subject to the conjecture that the base-b representation of $\sqrt{b}$ is disjunctive.

Before proceeding, we make a few remarks on the disjunctive-word hypothesis above. This assumption is implied by a conjecture of Borel (see the survey [44]) which says that for any integer $b \geq 2$ and real irrational algebraic number ζ, the digits of the base-b expansion of ζ form a normal word z. The strongest result towards this conjecture is due to Adamczewski and Bugeaud [1], who showed that the factor complexity function p_z of z satisfies

$$\liminf_{n \to \infty} \frac{p_z(n)}{n} = \infty.$$

We devote the rest of this section to the proof of Prop. 5.3.1. The key idea is as follows. Express n as $kd + r$. Then the largest number m such that $qm^d \leq pb^n$ is equal to $\left\lfloor \left(\sqrt[d]{pb^r/q} \right) b^k \right\rfloor$. This integer is obtained by shifting the radix point of the base-b representation of $\sqrt[d]{pb^r/q}$ by k places to the right.

Using this principle, it is easy to obtain the word y as the output of a transducer reading the word x and prove one reduction of the equivalence using Lem. 1.2.2. This transducer maintains a mod-b counter that is updated each time an occurrence of the predicate $\{qn^d \mid n \in \mathbb{N}\}$ is read, and a mod-d counter that is updated each time an occurrence of the predicate $\{pb^n \mid n \in \mathbb{N}\}$ is read. These counters are respectively initialised to $b - 1$ and $d - 1$.

The transducer prints its output as follows. In its first transition, it synchronises the components by printing the first $l - 1$ letters of y, i.e., all but the last digits before the radix point. Subsequently, let a be the value of the mod-b counter immediately after reading the $(kd + r)$-th letter that updates the mod-d counter. The r-th component of $y(k + l)$ is a.

The reduction of Acc_x to Acc_y closely mirrors the proof of Lem. 5.1.1: given an automaton $\mathcal{A}$, we can construct a transducer $\mathcal{B}$ such that the word $\mathcal{B}(y)$ is indistinguishable from x to $\mathcal{A}$. We therefore give only a sketch, and omit the full details of the technical bookkeeping.

Let us denote occurrences of $\{qn^d \mid n \in \mathbb{N}\}$ by the letter 1, those of $\{pb^n \mid n \in \mathbb{N}\}$ by the letter 2, and simultaneous occurrences by the letter 3. Note that the letter 3 can only occur if for some r, the number $\sqrt[d]{pb^r/q}$ is rational and has a denominator that divides some power of b — this can be decided, and is equivalent to the corresponding component of y having the suffix 0^ω.

The key idea gives us the following recurrence relation which defines how the transducer $\mathcal{B}$ works. Denote by $u_{r,k}$ the number of occurrences of $\{qn^d \mid n \in \mathbb{N}_{>0}\}$ that are not after pb^{kd+r}. We have

$$u_{r,k} = \sum_{i=0}^{l+k} b^k \cdot b^{l-i} \cdot y(i)(r)$$

and hence

$$u_{r,k+1} = b \cdot u_{r,k} + y(l + k + 1)(r).$$

We track this recurrence modulo the "period" m of the automaton $\mathcal{A}$ as defined for Lem. 5.1.1. Indeed, we can compute an index beyond which the gaps between consecutive occurrences of the letter 2 exceed the "threshold" $t = cm$ of $\mathcal{A}$. Beyond this index, when the transducer prints successive occurrences of the letter 2, it prints

$t + s$ intervening occurrences of 1 in $\mathcal{B}(y)$, where s is congruent modulo m to the actual number of intervening occurrences of 1 in the true order word x.

In this manner, $\mathcal{B}(y)$ is indistinguishable from x to $\mathcal{A}$, and we have thus reduced Acc_x to Acc_y. This completes the proof of Prop. 5.3.1.

5.4 Discussion

When asked for their opinion on why the decidability statuses of various MSO theories of expansions of $\langle \mathbb{N}; < \rangle$ withstand resolution, an expert in the automata theory community[4] said that these are typically questions concerning numbers rather than automata per se. Indeed, as we saw in the Introduction, the MSO theory of the structure $\langle \mathbb{N}; <, \{n^2 \mid n \in \mathbb{N}\}, \{n! \mid n \in \mathbb{N}\} \rangle$ subsumes the open Brocard-Ramanujan problem of determining whether the equation $n! + 1 = m^2$ has solutions other than $n \in \{4, 5, 7\}$. In the course of this chapter, we noted how the single predicate $\left\{ \binom{2n}{n} \mid n \in \mathbb{N}_{>0} \right\}$ is capable, via Kummer's theorem, of expressing an ensemble of open number-theoretic problems. We also saw in the Preliminaries that deciding the theory of $\langle \mathbb{N}; <, \mathsf{Primes} \rangle$ would resolve the Twin Prime Conjecture.

The reader might be aware of the results in [8], which include that the theory of $\langle \mathbb{N}; <, \mathsf{Primes} \rangle$ is decidable subject to Dickson's Conjecture, which is the linear case of Schinzel's Hypothesis H. Ingenious as the automata-theoretic arguments are, the result does not address the core number-theoretic difficulties — the conditional [8, Lem. 1] on which it hinges is a vast generalisation of the Twin Prime Conjecture.

Recently, [59] showed that the MSO theory of $\langle \mathbb{N}; <, U \rangle$ is decidable, where U is the set of positive values taken by a simple linear recurrence sequence with two dominant roots. The proof proceeds by showing that the characteristic word of this structure is obtained by transduction on an effective disjunctive word. Automata theory only constitutes the final step and is standard. However, the number theory required to invoke it is technically involved: e.g., the proof makes *several* appeals to Baker's theorem.

The point of this discussion is that the results of this chapter seem to have been conceived at a particularly opportune moment when the ingredients from number theory, symbolic dynamics, and automata theory were ripe, and the collaborators of [12] convened to synergise their respective expertise in each discipline. We hope

[4] Apologies, for I recall neither their identity, nor whether they would want the quote to be attributed to them in writing. The quote was reported to me verbally by my advisor Joël Ouaknine.

that interdisciplinary skill can continue wresting results from similarly challenging landscapes.

Concluding Perspective

We return to the broad question that we posed in the Introduction: how do automata run on shift spaces? The insights we have discovered in this thesis can be summarised as follows.

- If we regard automata as recognisers of languages, the congenial directive sequence is a natural device to factorise a word for the purpose of determining membership in an ω-regular language.
- If we view automata as string-to-string models of computation, i.e., sequential transducers, and seek to understand whether and how the output inherits structure from the input, then Krohn-Rhodes theory provides convenient footholds.

We are optimistic these observations will integrate into future theories. For instance, if one considers a class of words $\mathcal{W}$ (e.g., dendric words, multidimensional billiard words) and seeks to develop methods to parametrise $\mathcal{W} \cap L$ for a given ω-language L, it would suffice to develop a canonical system of directive sequences for $\mathcal{W}$. The success we witnessed for Sturmian words is in large part due to the intrinsic connections between continued fraction algorithms and return words to a letter, and it will be fascinating to discover and understand analogous connections for other classes $\mathcal{W}$.

Do we now understand how automata run on shift spaces?

Expressive Power and Preservation Theorems

Consider the following principle for compact minimal systems (Z, T) with invertible dynamics, established using the fact that a counter-free automaton is the cascade of reset automata. If x is the coding of the trajectory $(\zeta, T\zeta, T^2\zeta, \ldots)$ of a point ζ with respect to some finite partition $\mathcal{P}$ of the space Z, and $y = \mathcal{A}(x)$ for some counter-free automaton $\mathcal{A}$, then y is, modulo a finite prefix, also a coding of the trajectory of ζ under the same dynamics, albeit with respect to a finite partition $\mathcal{P}'$ that refines $\mathcal{P}$.

It suffices to show this for reset automata: indeed, use compactness and minimality to argue the uniform recurrence of x, find a return-time R for the occurrence of a reset letter, and use the invertibility of T to refine $\mathcal{P}$ based on the coding of the previous R steps.

The above argument mirrors the proofs of preservation of recurrence (Lem. 2.1.2) and admission of factor frequencies (Prop. 3.4.3, which assumes uniform recurrence) by counter-free automata, and can be used to show that the run of a counter-free automaton on a coding of a rotation is still, essentially, a more refined coding of the same rotation, or that the run of a counter-free automaton on a *Toeplitz word* has a Toeplitz suffix. (Recall that a word x is Toeplitz if for every index n, there exists a period p such that for all $k \geq 1$ we have $x(n) = x(n + p)$. Toeplitz words attract interest because their shifts are conjugate to a class of rotational systems called *odometers*. Consult, e.g., [5] for further reading, and the S-adic nature of Toeplitz words.)

Intuitively, taking the skew product of a shift with a counter-free automaton is much less disruptive than it is in general, in that it does not fundamentally alter the original dynamics. We have already seen this intuition in action: reset automata preserve the admission of factor frequencies for uniformly recurrent words (Prop. 3.4.3), but permutation automata do not (Lem. 3.2.2). A more elementary example of disruption comes from the failure of even the simplest permutation automaton to preserve the Toeplitz property.

Indeed, consider the Toeplitz period-doubling word $x \in \{0, 1\}^\omega$, whose n-th letter records the parity of $v_2(n + 1)$, i.e., the number of trailing 0-s in the binary representation of $n + 1$. We can run a two-state permutation automaton on x to obtain the Thue-Morse word $y \in \{0, 1\}^\omega$, whose n-th letter records the parity of $s_2(n)$, i.e., the sum of the binary digits of n. This construction is enabled by the binary carry identity, which gives the recurrence $s_2(n + 1) = s_2(n) - v_2(n + 1) + 1$. Thus, the automaton starts in state q_0, changes states upon reading 0, and stays in the same state upon reading 1. We get the Thue-Morse word as the sequence of states, which cannot have a Toeplitz suffix by [2, Fact 9].

In this thesis, we were nevertheless able to prove that permutation automata do preserve some properties such as recurrence (Lem. 2.1.3) and Boshernitzan's condition (Lem. 3.4.4). Curiously, we can readily prove that the run of a permutation automaton on the coding of an *interval exchange transformation (IET)* is the coding of an IET. (The space of an IET is the interval $[0, 1)$, which is partitioned into intervals $[a_i, b_i)$.

The transformation is a piecewise affine map of the form $x \mapsto x - c_i$ on the i-th interval, such that the overall map permutes the intervals — hence the name.)

Unlike the counter-free refinement principle, which keeps the dynamics unchanged, the preservation of IET origins uses a folkloric trick to keep track of the state of the permutation automaton $\mathcal{A}$ by creating copies of the dynamical system. The caveat is that this augmented dynamical system may not be minimal, i.e., the uniformly recurrent run which codes a trajectory may be dense in only a subset of the new space.

The meta question one may ask is as follows. Of those properties preserved by counter-free automata, why are only few also preserved by group automata, and hence arbitrary automata? This question concerns both the capability of automata to alter the original dynamics, and the very property whose preservation we seek to determine.

If we take expressive power as a proxy for the former, we can draw on a comprehensive understanding of this aspect. Recall that counter-free automata are precisely those that recognise (monadic) first-order definable languages [55], which in turn have a host of equivalent characterisations in terms of logic, regular expressions, and algebra [27]. It is now natural to investigate the significance of this coincidence on the first-order verification (and synthesis) of dynamical systems — given that running the relevant automata do not alter the dynamics, and barely change the return-time function, how can we profit by analysing the system ahead of time?

"Counter-free automata characterise a restricted fragment of ω-regular languages, and keep dynamics unperturbed" answers only part of our meta question. What makes preservation by group automata succeed for some properties, and fail for others? Are rotation and Toeplitz (and other properties that remind dynamical systems practitioners of *pure discrete spectrum*) too "rigid"? Is unique ergodicity too "malleable"? Did Boshernitzan's condition succeed because it is intrinsically "symmetric"? To make some of these notions precise for our question, do we even have the mathematical vocabulary yet?

We hope that we have conveyed, along with the questions, the inspiration to pursue them.

Bibliography

[1] Boris Adamczewski and Yann Bugeaud. On the Complexity of Algebraic Numbers I. Expansions in Integer Bases. *Ann. of Math.*, 165:547–565, 2007.

[2] Ibai Aedo, Uwe Grimm, Yasushi Nagai, and Petra Staynova. Monochromatic arithmetic progressions in binary Thue–Morse-like words. *Theoretical Computer Science*, 934:65–80, 2022.

[3] Jean-Paul Allouche and Jeffrey Shallit. *Automatic Sequences: Theory, Applications, Generalizations*. Cambridge University Press, 2003.

[4] Shaull Almagor, Toghrul Karimov, Edon Kelmendi, Joël Ouaknine, and James Worrell. Deciding omega-regular properties on linear recurrence sequences. *POPL*, 5, January 2021.

[5] Felipe Arbulú, Fabien Durand, and Bastián Espinoza. The Jacobs–Keane theorem from the $\mathcal{S}$-adic viewpoint. *Discrete and Continuous Dynamical Systems*, 44(10):3077–3108, 2024.

[6] Christel Baier, Florian Funke, Simon Jantsch, Toghrul Karimov, Engel Lefaucheux, Joël Ouaknine, David Purser, Markus A. Whiteland, and James Worrell. Parameter Synthesis for Parametric Probabilistic Dynamical Systems and Prefix-Independent Specifications. In *CONCUR 2022*, volume 243 of *LIPIcs*, pages 10:1–10:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

[7] Alan Baker. *Transcendental Number Theory*. Cambridge Mathematical Library. Cambridge University Press, 1975.

[8] Paul T. Bateman, Carl G. Jockusch, and Alan R. Woods. Decidability and Undecidability of Theories with a Predicate for the Primes. *J. Symbolic Logic*, 58:672–687, 1993.

[9] Heinz Bauer. *Measure and Integration Theory*. De Gruyter, Berlin, New York, 2001.

[10] Valérie Berthé and Vincent Delecroix. Beyond substitutive dynamical systems: S-adic expansions. *RIMS Kôkyûroku Bessatsu*, B46:81–123, 2014.

[11] Valérie Berthé, Charles Holton, and Luca Q. Zamboni. Initial powers of Sturmian sequences. *Acta Arithmetica*, 122(4):315–347, 2006.

[12] Valérie Berthé, Toghrul Karimov, Joris Nieuwveld, Joël Ouaknine, Mihir Vahanwala, and James Worrell. On the decidability of monadic second-order logic with arithmetic predicates. In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–14, 2024.

[13] Valérie Berthé, Toghrul Karimov, and Mihir Vahanwala. Automata on S-Adic Words. In *ICALP*, 2026.

[14] Valérie Berthé, Herman Goulet-Ouellet, Toghrul Karimov, Dominique Perrin, and Mihir Vahanwala. Preservation Theorems for Transducer Outputs, 2026.

[15] Valérie Berthé, Herman Goulet-Ouellet, Carl-Fredrik Nyberg Brodda, Dominique Perrin, and Karl Petersen. Density of group languages in shift spaces. *Ergodic Theory and Dynamical Systems*, 46(11):2645–2701, 2026.

[16] Valérie Berthé, Wolfgang Steiner, Jörg M. Thuswaldner, and Reem Yassawi. Recognizability for sequences of morphisms. *Ergodic Theory and Dynamical Systems*, 39(11):2896–2931, 2019.

[17] Michael D. Boshernitzan. A condition for unique ergodicity of minimal symbolic flows. *Ergodic Theory and Dynamical Systems*, 12(3):425–428, 1992.

[18] J. Richard Büchi. On a decision method in restricted second order arithmetic. In Ernest Nagel, Patrick Suppes, and Alfred Tarski, editors, *Logic, Methodology and Philosophy of Science: Proceedings of the 1960 International Congress*, pages 1–11, Stanford, California, 1962. Stanford University Press.

[19] Yann Bugeaud and Michel Laurent. Combinatorial structure of Sturmian words and continued fraction expansion of Sturmian numbers. *Annales de l'Institut Fourier*, 73(5):2029–2078, 2023.

[20] Olivier Carton and Wolfgang Thomas. The monadic theory of morphic infinite words and generalizations. *Information and Computation*, 176(1):51–65, 2002.

[21] Jon Chaika. Skew products over rotations with exotic properties. *Geometriae Dedicata*, 168(1):369–385, 2014.

[22] Thomas Colcombet. Green's Relations and Their Use in Automata Theory. In Adrian-Horia Dediu, Shunsuke Inenaga, and Carlos Martín-Vide, editors, *Language and Automata Theory and Applications*, pages 1–21, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[23] Thierry Combot. Computing Linear Relations between Polynomial Roots. *Mathematics of Computation*, 95(358):999–1022, 2026.

[24] David Damanik and Daniel Lenz. A condition of Boshernitzan and uniform convergence in the multiplicative ergodic theorem. *Duke Mathematical Journal*, 133(1):95 – 123, 2006.

[25] David Damanik and Daniel Lenz. Zero-measure Cantor spectrum for Schrödinger operators with low-complexity potentials. *Journal de Mathématiques Pures et Appliquées*, 85(5):671–686, 2006.

[26] Jean-Charles Delvenne, Petr Kůrka, and Vincent Blondel. Decidability and Universality in Symbolic Dynamical Systems. *Fundam. Inf.*, 74(4):463–490, December 2006.

[27] Volker Diekert and Paul Gastin. First-order definable languages. In Jörg Flum, Erich Grädel, and Thomas Wilke, editors, *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*, volume 2 of *Texts in Logic and Games*, pages 261–306. Amsterdam University Press, 2008.

[28] Tomasz Downarowicz and Benjamin Weiss. Lifting generic points. *Ergodic Theory and Dynamical Systems*, 44(9):2565–2580, 2024.

[29] F. Durand, B. Host, and C. Skau. Substitutional dynamical systems, Bratteli diagrams and dimension groups. *Ergodic Theory and Dynamical Systems*, 19(4):953–993, 1999.

[30] Fabien Durand. Linearly recurrent subshifts have a finite number of non-periodic subshift factors. *Ergodic Theory and Dynamical Systems*, 20, 08 2008.

[31] Fabien Durand. Decidability of the HD0L ultimate periodicity problem. *RAIRO - Theoretical Informatics and Applications*, 47, 11 2011.

[32] Fabien Durand. HD0L ω-equivalence and periodicity problems in the primitive case. *Unif. Distrib. Theory*, 7(1):199–215, 2012.

[33] Fabien Durand. Decidability of uniform recurrence of morphic sequences. *International Journal of Foundations of Computer Science*, 24(01):123–146, 2013.

[34] Fabien Durand and Dominique Perrin. *Dimension Groups and Dynamical Systems: Substitutions, Bratteli Diagrams and Cantor Systems*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2022.

[35] Calvin C. Elgot and Michael O. Rabin. Decidability and undecidability of extensions of second (first) order theory of (generalized) successor. *The Journal of Symbolic Logic*, 31(2):169–181, 1966.

[36] France Gheeraert, Herman Goulet-Ouellet, Julien Leroy, and Pierre Stas. Stability properties for subgroups generated by return words. *European Journal of Combinatorics*, 130:104224, 2025.

[37] France Gheeraert and Julien Leroy. $\mathcal{S}$-adic characterization of minimal dendric shifts. *Discrete Mathematics & Theoretical Computer Science*, vol. 27:2, Feb 2025.

[38] Amy Glen and Jacques Justin. Episturmian words: a survey. *RAIRO - Theoretical Informatics and Applications*, 43(3):403–442, 2009.

[39] Steven M. Gonek and Hugh L. Montgomery. Kronecker's approximation theorem. *Indagationes Mathematicae*, 27(2):506–523, 2016. In Memoriam J.G. Van der Corput (1890–1975) Part 2.

[40] Herman Goulet-Ouellet. Suffix-connected languages. *Theoretical Computer Science*, 923:126–143, 2022.

[41] Richard K. Guy. *Unsolved Problems in Number Theory*, volume 1 of *Problem Books in Mathematics*. Springer-Verlag, New York, 3rd edition, 2004.

[42] Jacques Hadamard. Les surfaces à courbures opposées et leurs lignes géodésiques. *Journal de Mathématiques Pures et Appliquées*, 4:27–73, 1898.

[43] Vesa Halava and Tero Harju. Undecidability of infinite post correspondence problem for instances of Size 9. *RAIRO - Theoretical Informatics and Applications*, 40(4):551–557, 2006.

[44] Glyn Harman. One Hundred Years of Normal Numbers. In *Surveys in Number Theory*, pages 57–74. A K Peters, 2002.

[45] Philipp Hieronymi, Dun Ma, Reed Oei, Luke Schaeffer, Christian Schulz, and Jeffrey Shallit. Decidability for Sturmian words. *Logical Methods in Computer Science*, Volume 20, Issue 3, Aug 2024.

[46] Juha Honkala. On the simplification of infinite morphic words. *Theoretical Computer Science*, 410(8):997–1000, 2009.

[47] Toghrul Karimov. *Algorithmic verification of linear dynamical systems.* PhD thesis, Universität des Saarlandes, 2023.

[48] Wolfgang Krieger. On unique ergodicity. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability*, volume 2, pages 327–346. Cambridge University Press, 1972.

[49] Kenneth Krohn and John Rhodes. Algebraic Theory of Machines. I. Prime Decomposition Theorem for Finite Semigroups and Machines. *Transactions of The American Mathematical Society - TRANS AMER MATH SOC*, 116, 04 1965.

[50] Serge Lang. *Introduction to Transcendental Numbers.* Addison-Wesley series in mathematics. Addison-Wesley Publishing Company, 1966.

[51] Mariusz Lemańczyk and Mieczysław K. Mentzen. Topological Ergodicity of Real Cocycles over Minimal Rotations. *Monatshefte für Mathematik*, 134(3):227–246, 2002.

[52] Douglas Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding.* Cambridge University Press, 1995.

[53] Angus Macintyre, A Wilkie, and P Odifreddi. On the decidability of the real exponential field. *Kreisel's Mathematics*, 115:451, 1996.

[54] Oded Maler. *On the Krohn-Rhodes Cascaded Decomposition Theorem*, pages 260–278. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[55] Robert McNaughton and Seymour A. Papert. *Counter-Free Automata (M.I.T. research monograph no. 65)*. The MIT Press, 1971.

[56] A.R. Meyer and C. Thompson. Remarks on algebraic decomposition of automata. *Mathematical Systems Theory*, 3:110–118, 1969.

[57] Hamoon Mousavi. Automatic Theorem Proving in Walnut, 2021.

[58] A. Muchnik, A. Semenov, and M. Ushakov. Almost periodic sequences. *Theoretical Computer Science*, 304(1):1–33, 2003.

[59] Joris Nieuwveld and Joël Ouaknine. On Expansions of Monadic Second-Order Logic with Dynamical Predicates. In *MFCS*, pages 80:1–80:17, 2025.

[60] Joris Adriaan Nieuwveld. *Algorithmic problems for linear recurrence sequences*. PhD thesis, Universität des Saarlandes, 2025.

[61] Reed Oei, Dun Ma, Christian Schulz, and Philipp Hieronymi. Pecan: An Automated Theorem Prover for Automatic Sequences using Büchi Automata, 2021.

[62] Joël Ouaknine and James Worrell. Positivity problems for low-order linear recurrence sequences. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, page 366–379, USA, 2014. Society for Industrial and Applied Mathematics.

[63] Dominique Perrin and Jean-Éric Pin. Semigroups and automata on infinite words. In *Semigroups, Formal Languages and Groups*, pages 49–72. Kluwer Academic Publishers, 1995.

[64] Nir Piterman. From Nondeterministic Büchi and Streett Automata to Deterministic Parity Automata. *Logical Methods in Computer Science*, Volume 3, Issue 3, Aug 2007.

[65] Yu. L. Pritykin. Finite-automaton transformations of strictly almost-periodic sequences. *Mathematical Notes*, 80(5):710–714, 2006.

[66] Martine Queffélec. *Substitution Dynamical Systems - Spectral Analysis*, volume 1294 of *Lecture Notes in Mathematics*. Springer, Berlin, Heidelberg, 2nd edition, 2010.

[67] Andrzej Schinzel and Robert Tijdeman. On the equation $y^m = p(x)$. *Acta Arithmetica*, 31:199–204, 1976.

[68] A. Semenov. Logical theories of one-place functions on the set of natural numbers. *Mathematics of the USSR-Izvestiya*, 22(3):587, 1984.

[69] Jeffrey Shallit. *The Logical Approach to Automatic Sequences: Exploring Combinatorics on Words with Walnut*. London Mathematical Society Lecture Note Series. Cambridge University Press, 2022.

[70] Michel Waldschmidt. *Diophantine Approximation on Linear Algebraic Groups*. Springer, 2000.

[71] Thomas Wilke. An Eilenberg theorem for ω-languages. In *Automata, Languages and Programming*, pages 588–599, 1991.

Index

algebraic number, 91
  degree, 91
  height, 91
  minimal polynomial, 91
almost-periodicity, 21
  effectively almost-periodic, 22
alphabet, 7
Arnoux-Rauzy words, 19
automaton, 28
  deterministic automaton, 27
  nondeterministic automaton, 28

Birkhoff average, 40
Borel set, 39

Champernowne word, 19
characteristic word of MSO structure, 28
cobounding map, 69
computable factor frequencies, 8
congenial augmented directive sequence, 24
continued fraction expansion, 13
  convergents, 14
  locally best approximants, 14
counter-free automaton, 35
cutting sequence, 10
  gradient, 10
cylinder, 39

dendric words, 80
directive sequence, 12
  centred representation, 76
  generating sequence, 12
    congenially generated, 24
  partial quotient, 80
  primitive directive sequence, 12
  strongly primitive directive sequence, 12
disjunctive word, 19

effective word, 7
effectively procyclic, 89
effectively sparse set of predicates, 89

factor frequency, 8
factor language, 8
factor of word, 8
Fibonacci word, 9

generic point, 40
golden shift, 38
group, 32
group automaton, 35

interval exchange transformation (IET), 107

letters, 7
linear recurrence, 20
linear recurrence sequence, 93

dominant root, 93
simple root, 93
liveness property, 47

measure, 39
ergodic measure, 58
invariant measure, 39
uniform invariant measure, 40
Monadic Second-Order Logic (MSO), 28
monoid, 32
morphic word, 9
primitive morphic word, 9
morphism, 33
MSO sentence, 29
MSO theory, 29
decidable MSO theory, 29

normal word, 19

occurrence in word, 8
odometer, 107
omega-regular language, 29
Ostrowski numeration, 14

permutation automaton, 35
permutation-reset automaton, 35
prefix-independent language, 47
product topology, 7
pure discrete spectrum, 108

recurrent factor, 19
recurrent word, 19
effectively recurrent word, 31
regular language, 27
reset automaton, 35
return word, 19
recurrent return word, 19
return group, 71
return-time function, 20
run of automaton on word, 28

S-adic word, 12
semigroup, 33
omega-semigroup, 33
shift (space), 38
forbidden factors, 38
generated shift, 38
minimal shift, 38
shift operator, 37
Sturmian shift, 38
typical words, 38
shift of finite type (SFT), 38
shift space
full shift, 38
skew product, 60
sofic shift, 38
Sturmian substitutions, 14
Sturmian word
characteristic Sturmian word, 9
formal intercept, 15
intercept, 10
lower mechanical word, 10
slope, 9
upper mechanical word, 10
substitution, 8
left-proper substitution, 75
non-erasing substitution, 8
primitive substitution, 8
substitutive word, 8
primitive substitutive word, 8

Toeplitz, 107
toric word, 10
torus generated by translation, 96

transducer, 28
    non-erasing transducer, 28
transition monoid, 33
Tribonacci word, 19

uniform factor frequencies, 58
uniform recurrence, 20
    effectively uniformly recurrent, 20
unique ergodicity, 58

volatility function, 21

window function, 21
word, 7
word length, 7