UNIVERSIDAD DEL BÍO-BÍO

Facultad de Ciencias Empresariales
Departamento de Ciencias de la Computación y
Tecnologías de la Información

An Aperiodic, Complete, Reversible and 2-Symbol Turing Machine.

Por
Pablo Alfonso Concha Vega

Tesis para optar al grado de Magíster
en Ciencias de la Computación

Dirigido por:
Dr. Rodrigo Torres
Universidad del Bío-Bío, Chillán, Chile

2019 - II

A mi madre y a mi padre.

Agradecimientos

Quisiera comenzar agradeciendo a mi madre y a mi padre, para quienes siempre he sido primerísima prioridad y quienes se han preocupado de brindar las condiciones necesarias para mi formación, ya que sin ellos lograr llegar a esta instancia hubiera sido muchísimo más complicado. Además, quiero reconocer el apoyo de otros miembros de mi familia, en particular el de mis abuelos maternos.

Quiero también agradecer a todos aquellos docentes que formaron parte de mi desarrollo académico, en especial al profesor Rodrigo quien, además de adentrarme en diversos tópicos de investigación, siempre ha tenido una alta disponibilidad y paciencia con mi persona.

Asímismo, deseo agradecer a mi novia y amigos quienes ayudaron a que esta etapa de mi vida estuviera llena de buenos momentos.

Finalmente, agradezco a ANID por apoyar económicamente este trabajo.

Resumen

Las máquinas de Turing han sido estudiadas como sistemas dinámicos por más de dos décadas, inicialmente por Moore, pero posteriormente formalizadas por Kůrka, quien propuso un sistema dinámico llamado *máquina de Turing con cabezal móvil*. Él también conjeturó que *toda máquina de Turing tiene al menos un punto periódico en el modelo de cabezal móvil*, esto quiere decir que cada máquina bajo este modelo tiene al menos una configuración que se repite en el proceso de computación. Sin embargo, actualmente ya existen algunos ejemplos de máquinas de Turing que, de hecho, no tienen ningún punto periódico, mostrando que la conjetura de Kůrka era incorrecta. Es más, una de estas máquinas, apodada SMART, tiene otras propiedades interesantes como lo son la reversibilidad y la minimalidad topológica, por ejemplo. Esta máquina tiene cuatro estados y funciona con un alfabeto de tres símbolos.

En esta tesis estudiamos las propiedades dinámicas de BinSmart, que es la primera máquina de Turing que, además de ser aperiódica y reversible, funciona con un alfabeto de dos símbolos. Esta máquina resulta ser topologicamente minimal (por lo tanto transitiva) y no simétrica en el tiempo. También estudiamos su t-shift, que es un subshift que se deriva del modelo de máquina de Turing con cabezal móvil, el cual almacena información importante del historial de computación de la máquina. Finalmente, demostramos que el t-shift de BinSmart es un subshift substitutivo.

Abstract

Turing machine has been studied as a dynamical system for more than two decades, initially by Moore, but later formalized by Kůrka, proposing a dynamical system named *Turing machine with moving tape*. He also conjectured that *every Turing machine has at least one periodic point in the 'Turing machine with moving tape' model*, it means that every machine in this model has at least one configuration that is repeated in the evolution of the computation process. Nevertheless, nowadays there are some examples of Turing machines that, in fact, do not have any periodic point, showing that the Kůrka's conjecture was wrong. Moreover, one of these machines, named SMART, has other interesting properties like reversibility and topological minimality, for example. This machine has four states and works over an alphabet of three symbols.

In this thesis, we study the dynamical properties of BinSmart, the first aperiodic, reversible, and 2-symbols Turing machine. This machine results to be topologically minimal (therefore transitive) but not time-symmetrical. We also study its t-shift, which is a subshift that derives from the Turing machine with moving tape model that contains important information of the computation history of the machine. We prove that this t-shift is a substitutive subshift.

Contents

Part I

Introduction

Chapter 1

Introduction.

1.1 Introduction and Motivation

The term *computation* refers to any kind of mathematical calculation that includes both arithmetical and non-arithmetical steps and follows a well-defined model, for example, an algorithm. While algorithms had an important role in certain areas of mathematics, prior to the 1930's they had not been studied as mathematical objects by themselves. Alan Turing [12] changed this by introducing a type of imaginary machine which could take an input and process it with a finite number of steps until getting the final output. This mathematical model of computation defines an abstract machine, which classically consists of a head that, following some defined instructions, can manipulate symbols and move along an infinite tape (but with a finite input). The part of the tape that is not used in the process of computation is filled with a special symbol usually known as *blank symbol*. This model was developed as a formalization of the concept of computation.

The study of Turing machine is mainly focused on defining computability and complexity, that in simple words means to study whether computers can solve a certain problem and how much space or time is necessary for solving this problem. One of the most studied problems related with this topic is the *halting problem*, which consists in determining, given a computer program and an input, whether the program will finish running or will be running forever. Alan Turing proved in 1936 that an algorithm to solve the halting problem for all possible input (A program and an input for this program) cannot exists.

Nevertheless, in this thesis, we do not study either the classical Turing machine model or computability and complexity. Instead, we look Turing machine as a *symbolic system*, more specifically, a model defined by Kůrka [7] in 1997 named *Turing machine with moving tape*, which, as its name says, it is the tape the component that has the ability to move while the head keeps stationary. It is important to remark that if the model is not defined this way, it would not be compact, which is a serious drawback in topological dynamics [7]. The idea behind this model is to study the dynamics of Turing machine, so we do not consider either initial or final states, since the computation may start in any state and proceed infinitely. We do not need to restrict the computation only to finite tape contents either, because the computation is carried out over an arbitrarily long part of the tape. Also, the blank symbol becomes part of the alphabet because without doing this we can not correctly define the metric space. Simply stated, the Turing machine with moving tape model can be defined with a finite set of inner states, a finite alphabet, and a transition function.

Considering this dynamical point of view, interesting questions emerge when we work over infinite 'inputs' or infinite computational time. Can we predict the behavior of the computer? Can we know if the computer will reach a specific state of computation? Can we determine if the computation will fall into a loop? if it does not, will it reach any possible state of computation? How much the dynamical point of view can tell us about computation? Some of these questions are already solved [4], but these are the type of questions we analyze in this thesis with the help of a really special Turing machine nicknamed 'BinSmart', which is the first Turing machine known for having an aperiodic behavior over a binary alphabet.

1.2 Investigation Objectives

The objectives of this thesis are to demonstrate whether the BinSmart Turing machine satisfies some dynamical properties and to prove if its derived t-shift is a substitutive subshift or not. More specifically, the properties of interest are the following ones:

- Aperiodicity
- Topological transitivity
- Topological minimality
- Time-symmetry
- Substitutivity

We will properly define them in chapter 2. In the course of this work, we will see that the BinSmart machine satisfies all of these properties except time-symmetry.

Part II

Context

Chapter 2

Definitions.

2.1 Dynamical System

A *dynamical system* is a pair (X, T), where X is a compact metric space and $T : X \to X$ a continuous self-map called *global transition function*. The compact metric space evolves in time through the global transition function, thus the nth evolution of an element $x \in X$ is denoted by $T^n(x)$. When the dynamical system works over discrete time it is called *discrete dynamical system*, and when the metric space is defined in a discrete way, it is called a *symbolic system*.

2.1.1 Metric space

Many spaces have a natural way to measure the distance between pairs of points, for example, we can easily get the distance between $x, y \in \mathbb{R}$ using $d(x, y) = |x - y|$. Moreover, this metric (or distance function) can be generalized for any euclidean n-dimensional space, but there are also more exotic examples of interest for mathematicians. One of them is the *discrete metric*, which assigns a 0 to the distance from a point to itself and a value 1 to the distance from a point to any other. Thus, a metric generalizes the usual concept of distance to more general settings. Furthermore, a specific metric has a big impact when a topology is defined, as we will see later.

Following [9], a *metric space* (X, d) is a set X together with a metric $d : X \times X \to [0, \infty)$ such that, for all points $x, y, z \in X$,

1. $d(x, y) = 0$ if and only if $x = y$,
2. $d(x, y) = d(y, x)$,
3. $d(x, z) \leq d(x, y) + d(y, z)$.

2.1.2 Orbit

In a dynamical system (X, T), the orbit $\mathcal{O}(x)$ of a point $x \in X$ is defined by $\mathcal{O}(x) = (T^n(x))_{n \in \mathbb{N}}$, in other words, the orbit of a point contains all its evolutions.

Pre-periodic and periodic orbits: In a dynamical system (X, T) a point x has a *periodic orbit* if there exists $n \in \mathbb{N}$ such that $x = T^n(x)$. The point $x \in X$ is called *periodic point* with period n. If $n = 1$, x is a *fixed point.*

If there exists two different numbers $n, m \in \mathbb{N}$ such that $T^n(x) = T^m(x)$, then $\mathcal{O}(x)$ is called *pre-periodic orbit*, in which case $x \in X$ is called *pre-periodic point.*

Aperiodic orbit: In a dynamical system (X, T), an orbit is *aperiodic* if no point is repeated. Formally, given a $x \in X$, the orbit $\mathcal{O}(x)$ is aperiodic if there does not exists a number $n > 0$ such that $T^n(x) = x$.

2.1.3 Subshifts, languages and words

There exists a specific type of symbolic system called *subshift*, which is based in a space of words evolving through the shift function. To give a formal definition, we need to provide some definitions about words.

Given a finite set Σ, called *alphabet*, $\Sigma^{\mathbb{Z}}$ is the set of bi-infinite sequences of elements of Σ, called *bi-infinite words.* Σ^{ω} (${}^{\omega}\Sigma$) represent the set of right (left) infinite sequences of elements of Σ, called *infinite words to the right (left).* The set of infinite words to the right can be also represented by $\Sigma^{\mathbb{N}}$. Finally, Σ^* represents the set of finite concatenations of elements of Σ, called *finite words*, including the word of length 0; the empty word ϵ. Two finite words $v = v_0...v_n$ and $v' = v'_0...v'_n$ can be *concatenated* just by putting them together: $vv' = v_0...v_nv'_0...v'_n$. We can also concatenate a finite word v with a right infinite word u: $vu = v_0...v_nu_0u_1....$ A finite word v is said to be a subword of another (finite or infinite) word u, if there exists two indexes $i < j$, such that $u_iu_{i+1}...u_j = v$. This is denoted by $v \sqsubseteq u$ (and $u \sqsupseteq v$). If the index i is equal to 0, we say that v is a *prefix* of u and is denoted by $v \sqsubset_p u$ (and $u \sqsupset_p v$). If the index j is equal to l, where l is the length of u, we say that v is a *suffix* of u and is denoted by $v \sqsubset_s u$ (and $u \sqsupset_s v$). It is important to say that this definition of subword may vary in other works.

Now, let us introduce the *shift* function σ, which is defined both in $\Sigma^{\mathbb{Z}}$ and $\Sigma^{\mathbb{N}}$ by $\sigma(u)_i = u_{i+1}$. Following [11], given a subset $S \subseteq \Sigma^{\mathbb{Z}}$ (or $S \subseteq \Sigma^{\mathbb{N}}$), a formal language is defined:

$$\mathcal{L}(S) = \{z \in \Sigma^* | (\exists w \in S) z \sqsubseteq w\}$$

Reciprocally, given a formal language L, a set of infinite sequences can be defined in $\Sigma^{\mathbb{M}}$ ($\mathbb{M} \in \{\mathbb{Z}, \mathbb{N}\}$):

$$S_L = \{w \in \Sigma^{\mathbb{M}} | (\forall z \sqsubseteq w) z \in L\}$$

When S satisfies $S_{\mathcal{L}(S)} = S$, it is called a *subshift.*

2.1.4 Cantor metric

The metric spaces studied in this thesis use the *Cantor metric* [8], that can be defined for bi-infinite words $u, v \in \Sigma^{\mathbb{Z}}$ as

$$d(u,v) = 2^{-i}, \text{where } i = min\{|j| : u_j \neq v_j, j \in \mathbb{Z}\}$$

and for finite words $s, t \in \Sigma^{\mathbb{N}}$ as

$$d(s,t) = 2^{-i}, \text{where } i = min\{j : s_j \neq t_j, j \in \mathbb{N}\}$$

I.e, the Cantor metric is simply 2 to the negative i power, where i is the first position of discrepancy between two words.

2.2 Topology

Before defining what a topology is, we need to present some concepts, so let us define them first.

Given a metric space (X,d), we can define a *ball* as $B_r(x) = \{y \in X : d(x,y) < r\}$. Also, using this definition, we can define an *open set* as subset $U \subset X$ such that for every point $x \in U$ there exists a ball $B_r(x)$ for some $r > 0$ that completely belongs to U. We can also define the *closure* of a subset $U \subset X$ as $\overline{U} = \{x \in X : \forall r > 0, B_r(x) \cap U \neq \emptyset\}$.

Now, let us define topology. Let X be a set. Let Ω be a collection of its subsets such that:

1. the union of any collection of sets that are elements of Ω belongs to Ω,
2. the intersection of any finite collection of sets that are elements of Ω belongs to Ω,
3. the empty set $\emptyset$ and the whole X belong to Ω.

Then, as [13] states:

- Ω is a *topology* on X,
- the pair (X,Ω) is a *topological space*,
- elements of X are points of this topological space,
- elements of Ω are open sets of the topological space (X,Ω).

2.2.1 Topological dynamical system

A *topological dynamical system* (X,T,Ω) is a dynamical system (X,T) with a topology Ω such that T is continuous. When we refer to a topological dynamical system, we omit the collection Ω, which is defined by the open sets of the metric $d : X \times X \to \mathbb{N}$ in our cases, by the *balls* B_x as $\Omega = \{B_r(x) : x \in X, r > 0\}$.

2.2.2 Perfect set

A point $x \in S$, where $S \subset X$ is called an *isolated point* if there exists a neighborhood of x that does not contain any other point of S. Thus, a subset of a topological space is called a *perfect set* if it has no isolated points. In other words, given a perfect set S , any point can be approximated arbitrarily well by other points from the set.

2.3 Turing Machine

A Turing machine (TM) is a computational model that describes an abstract machine, which consists of a *head* that reads symbols from a *tape*, which are modified following some previously defined instructions of the machine. Turing machines are mainly used to define computability (can this problem be solved by a computer?) and complexity (how much space/time is necessary for a machine to solve this problem?). In this work, the dynamics of Turing machines are studied, therefore some preliminary considerations have to be made in order to specify, in the best way, Turing machines in the context of dynamical systems.

Formally, a Turing machine M is a tuple (Q, Σ, δ), where Q is a finite set of states, Σ is a finite set of symbols (a finite alphabet) and $\delta \subseteq Q \times \Sigma \times Q \times \Sigma \times \{-1, 0, +1\}$ is the writing/moving relation of the machine.

2.3.1 Configuration of a Turing machine

A TM works over a tape, usually bi-infinite, full of symbols from Σ.

A *configuration* is an element $(r, i, w) \in Q \times \mathbb{Z} \times \Sigma^{\mathbb{Z}}$. A *finite configuration* is an element $(r, i, v) \in Q \times \{0, 1, ..., m-1\} \times \Sigma^m$ for some $m \in \mathbb{N}$. A *right semi-infinite configuration* is an element $(r, i, u) \in Q \times \mathbb{N} \times \Sigma^{\omega}$. A *left semi-infinite configuration* is an element $(r, i, u) \in Q \times (-\mathbb{N}) \times {}^{\omega}\Sigma$.

2.3.2 Instructions of a Turing machine

Basically, an *instruction* is what takes the machine from one configuration to another, in other words, instructions define the behavior of Turing machines, and particularly, the evolution of configurations.

Mathematically, an instruction is a quintuple $(r, s, s', r', c) \in Q \times \Sigma \times \Sigma \times Q \times \{-1, 0, +1\}$ and it can be applied to a configuration (r'', i, w) if $w_i = s$ and $r'' = r$, leading to the configuration $(r', i + c, w')$, where $w'_i = s'$ and $w'_k = w_k$ for all $k \neq i$. If the configuration is finite or (right or left) semi-finite with domain K, and $i + c \notin K$, then the instruction cannot be applied and the machine halts. If no instruction can be applied, the machine halts too.

Definition 2.1. *For configurations x, y, we say that a Turing machine M reaches configuration y from x if y is the result of evolving M on x over a finite number of steps, and we denote this by $x \vdash^* y$. This notation is considered for finite, semi-finite and infinite configurations.*

In the computational context, Turing machines have an initial and a final state, but since we are studying its dynamics, we omit these definitions.

2.3.3 Deterministic Turing machine

A Turing machine M is *deterministic* if for any configuration (r, i, w), at most one instruction can be applied. This is equivalent to give δ as a (possibly partial) function $\delta : Q \times \Sigma \to Q \times \Sigma \times \{-1, 0, +1\}$.

2.3.4 Complete Turing machine

A Turing machine M is *complete* if for each configuration (r, i, w), at least one instruction can be applied. This is equivalent to say that the machine never halts.

2.3.5 Injective Turing machine

A Turing machine M is *injective* when δ is injective. This is:

$$(\forall q, q' \in Q)(\forall s, s' \in \Sigma) : \delta(q, s) = \delta(q', s') \Rightarrow q = q' \wedge s = s'$$

This is equivalent to say that every configuration comes from at most one pre-image or previous configuration.

2.3.6 Reversible Turing machine

A Turing Machine M is *reversible* if it is deterministic and injective.

In this work, we will use the definition of reverse Turing machine from [3]. A reversible Turing machine can be characterized by a pair (ρ, μ), where $\rho : Q \times \Sigma \to Q \times \Sigma$ is a partial injective function and $\mu : Q \to \{-1, 0, +1\}$ is a partial function, such that δ is characterized by all the instructions of the form $(r, s, s', r', \mu(r'))$ where $r \in Q$, $s \in \Sigma$ and $(r', s') = \rho(r, s)$.

Indeed, the movement portion of the instructions depends on the state at which it goes in a reversible Turing machine; if not, the Turing machine has a configuration with more than one pre-image, and therefore it would be not injective. Now we can define the following:

Definition 2.2. *The reverse of a Turing machine $M = (Q, \Sigma, \delta)$ is defined by $M^{-1} = (Q, \Sigma, \delta^{-1})$, where $(r', s', s, r, -\mu(r)) \in \delta^{-1}$ if and only if $r \in Q$, $s \in \Sigma$ and $(r', s') = \rho(r, s)$.*

The reverse machine is called this way because it reverses the computation. In this case, we need to define the function $\phi : (\Sigma^{\mathbb{Z}}, \mathbb{Z}, Q) \to (\Sigma^{\mathbb{Z}}, \mathbb{Z}, Q)$ as $\phi(w, i, r) = (w, i - \mu(r), r)$, then the reverse computation is obtained by applying $\phi^{-1} \circ M^{-1} \circ \phi$.

2.4 Turing Machine as a Dynamical System

Dynamical systems of Turing machine is a paradigm formalized by Kůrka [7] (but firstly introduced by Moore [10]) and it gives us a strong tool for the study of the dynamics of Turing machine. In this thesis, we will consider the dynamical system called *Turing machine with Moving Tape* (TMT), which consists in putting the head at the center of the tape (the 0 position) and only moving the tape instead.

The dynamical system (X, T) for TMT consists in: $X \subseteq {}^{\omega}\Sigma \times Q \times \Sigma^{\omega}$ and $T : X \to X$ is the application of δ by moving the tape instead of the head. An element from ${}^{\omega}\Sigma \times Q \times \Sigma^{\omega}$ is called a *TMT configuration*.

To have a better understanding of how this system works, let us specify the way that instructions are applied: instruction (r, u_0, s', r', c) is applied to a TMT configuration $(...w_2 w_1 w_0, r, u_0 u_1 u_2...)$ resulting in:

- if $c = -1$, $(...w_3w_2w_1, r', w_0s'u_1...)$
- if $c = 0$, $(...w_2w_1w_0, r', s'u_1u_2...)$
- if $c = +1$, $(...w_1w_0s', r', u_1u_2u_3...)$

We call a *finite configuration* of TMT, a tuple $(v, r, v') \in {}^*\Sigma \times Q \times \Sigma^*$. Additionally, the metric used to measure the distance between a pair of points $(w, r, u), (w', r', u') \in X$ is:

$$d((w,r,u),(w',r',u')) = \begin{cases} 1 & \text{if } r \neq r' \\ 2^{-i} & \text{in other cases} \end{cases} \text{,where } i = min\{j : w_{-j} \neq w'_{-j} \vee u_j \neq u'_j, j \in \mathbb{N}\}$$

As we can see, this is a modification of the Cantor metric.

2.4.1 The t-shift

Taking into account the TMT dynamical system, we can define the projection $\pi : X \to Q \times \Sigma$ by $\pi(w, r, w') = (r, w'_0)$.

The *t-shift*, denoted by $S_T \subseteq (Q \times \Sigma)^{\mathbb{N}}$, is the sets of orbits $\tau(x) = (\pi(T^n(x)))_{n\in\mathbb{N}}$ for $x \in X$. In other terms, $S_T = \{\tau(x) : x \in X\}$.

In figure 2.1, a comparison among the classic Turing machine model, TMT and t-shift can be seen through an example.

$$\begin{array}{cccccc} 1&0&1&1&1&0\\ &&q_0&&&\\ 1&0&0&1&1&0\\ &q_1&&&&\\ 1&0&0&1&1&0\\ &&q_0&&&\\ 1&0&1&1&1&0\\ &&&q_2&&\\ &&...&&& \end{array} \qquad \begin{array}{cccccc} 1&0&1&1&1&0\\ &&q_0&&&\\ 0&1&0&0&1&1\\ &&q_1&&&\\ 1&0&0&1&1&0\\ &&q_0&&&\\ 0&1&1&1&0&0\\ &&q_2&&&\\ &&...&&& \end{array} \qquad \begin{array}{ccccc} 1&0&0&1&...\\ q_0&q_1&q_0&q_2&\\ 0&0&1&1&...\\ q_1&q_0&q_2&q_2&\\ 0&1&1&0&...\\ q_0&q_2&q_2&q_2&\\ 1&1&0&0&...\\ q_2&q_2&q_2&q_2&\\ &&...&& \end{array}$$

Turing machine model TMT model t-shift

Figure 2.1: Examples of the evolution of a TM in its dynamical models (the classic one, TMT and t-shift). The represented machine has the instructions $(q_0, 0, 1, q_2, +1)$, $(q_0, 1, 0, q_1, -1)$, $(q_1, 0, 0, q_0, +1)$ and $(q_2, \alpha, \alpha, q_2, +1)$, $\forall\alpha \in \Sigma$.

As we can see, the t-shift stores the read symbol and the current inner state at each step of computation for every possible configuration i.e. it contains relevant information of the computation history of the machine.

2.4.2 Cylinder

In this context, a ball is called a *cylinder* and it is defined by $B_r(x) = \{y \in {}^\omega\Sigma \times Q \times \Sigma^\omega : d(x, y) < r\}$, but we can also define it in a more intuitive way: Given a finite configuration $(v, r, v') \in \Sigma^* \times Q \times \Sigma^*$, its cylinder is:

$$[v, r, v'] = \{(w, r', w') \in {}^{\omega}\Sigma \times Q \times \Sigma^{\omega} : (r = r')(w \sqsupset_s v)(w' \sqsupset_p v')\}$$

2.4.3 Turing machine dynamical properties

Here, we present the properties linked with the Turing machine dynamical system that we will study.

Aperiodicity

The first of this properties is *aperiodicity* which says that a dynamical system does not have any periodic point. In the Turing machine context, we will call *periodic configuration* instead of periodic point, then a Turing machine is aperiodic if it does not have any periodic configuration. In mathematical terms:

$$(\forall x \in X)(\forall n > 0) : T^n(x) \neq x$$

Topological transitivity

A dynamical system (X, T) is topologically transitive if there exists a point $x \in X$ such that for all point $y \in X$ we got that $y \in \mathcal{O}(x)$. When this happens, we say that x is a *transitive point* and $\mathcal{O}(x)$ is *dense*.

Now, let us contextualize this property for the TMT dynamical system.

Definition 2.3. *Let (X, T) be a TMT dynamical system. (X, T) is topologically transitive if*

$$(\forall u, v \in \Sigma^* \times Q \times \Sigma^*)(\exists x \in [u])(\exists n > 0) : T^n(x) \in [v]$$

In TMT, *topological transitivity* means that there exists at least one TMT configuration that can reach any TMT finite configuration, so in this particular case, we can equivalently define topological transitivity as follows:

$$(\exists x \in X)(\forall u \in \Sigma^* \times Q \times \Sigma^*)(\exists n > 0) : T^n(x) \in [u]$$

These definitions are equivalent just because we work over a perfect set [1].

Topological minimality

If every point of a dynamical system is topologically transitive, then it is topologically minimal.

Definition 2.4. *Let (X, T) be a TMT dynamical system. (X, T) is topologically minimal if*

$$(\forall u, v \in \Sigma^* \times Q \times \Sigma^*)(\forall x \in [u])(\exists n > 0) : T^n(x) \in [v]$$

Similar to topological transitivity, topological minimality implies that every TMT configuration reaches every TMT finite configuration. This is equivalent to:

$$(\forall x \in X)(\forall u \in \Sigma^* \times Q \times \Sigma^*)(\exists n > 0) : T^n(x) \in [u]$$

Just as before, this equivalency is possible because we are working over a perfect set.

Time-symmetry

Time-symmetry is a property firstly studied in physical systems and it is considered stronger than reversibility. When a system presents this property, it is indistinguishable when the system goes forward or backward in time. The definition for this property used in this thesis has been taken from [3].

Definition 2.5. *A reversible Turing machine $M = (Q, \Sigma, \delta)$ is said to be time-symmetric if there exists involutions $h_Q : Q \to Q$ and $h_\Sigma : \Sigma \to \Sigma$ such that:*

$$(h_Q(r), h_\Sigma(s), h_\Sigma(s'), h_Q(r'), c) \in \delta^{-1} \Leftrightarrow (r, s, s', r', c) \in \delta$$

Substitutive subshift

A *substitution* is a morphism $\phi : \Sigma^* \to \Sigma^*$, which can be extended to $\Sigma^{\mathbb{N}}$. A fixed point of ϕ is a word $w \in \Sigma^{\mathbb{N}}$ such that $\phi(w) = w$. A subshift is substitutive if it is the closure of the orbit of a fixed point of some substitution. In that case, we can define the subshift with that substitution.

Chapter 3

Related Work.

3.1 Kůrka's Conjecture

As we mentioned before, Kůrka defined the TMT model, being the first one to formalize Turing machine as a dynamical system in his article 'On topological dynamics of Turing machines' [7], which in addition to be a seminal work and to obviously have some interesting results, contains the following conjecture.

Conjecture 3.1 ([7])**.** *every TMT has a periodic point.*

3.2 The First Counterexample

Nevertheless, the authors Blondel, Cassaigne, and Nichitiu [2], developed a Turing machine inspired in the 'bounded searches' of Hooper [5], this was the very first time that this type of searches was applied in a complete Turing machine. This machine has an aperiodic behavior in the TMT model, proving that the Kůrka's Conjecture was wrong. Their main machine has 36 states and works over 4 symbols but they also proposed a smaller machine with only 6 states and the same alphabet.

To understand their result, let us show the machine K_0 (Figure 3.1). This machine would be the obvious but naive first attempt to develop an aperiodic Turing machine. Its behavior consists in applying simple searches of a 1 symbol in the tape, either to the right with the state A, or to the left with the state C. As we can see, this machine does not have any periodic point except when the tape is full of 0s and the machine is in one of its search states, in those cases, we immediately see a periodic behavior, a fixed point. Moreover, any machine with an instruction of the form $(r, \alpha, \alpha, r, c)$, i.e. with a search state r, has at least one periodic configuration.

At this point, the authors introduce the 'bounded searches', that basically consists in applying a search of a specific symbol only in a specific portion of the tape and, if the search fails, the machine will start a 'wider' search. On the other hand, if the search does not fail; ergo, it successfully finds the necessary symbol, the machine will 'move' it to the right(left) if it is under a right(left) search. They use this concept to develop the K_1 Turing machine, that searches a positive number, from its alphabet $\Sigma = \{0, 1, 2, 3\}$, only in three cells before making a recursive call. Since this machine is considerably big, we will not show it here, instead, we can see the K_2 Turing machine (figure 3.2), also proposed by Blondel et al., which is a minimized version of the K_1 machine generated by reducing the steps used in the bounded searches and merging some states that have a similar behavior.

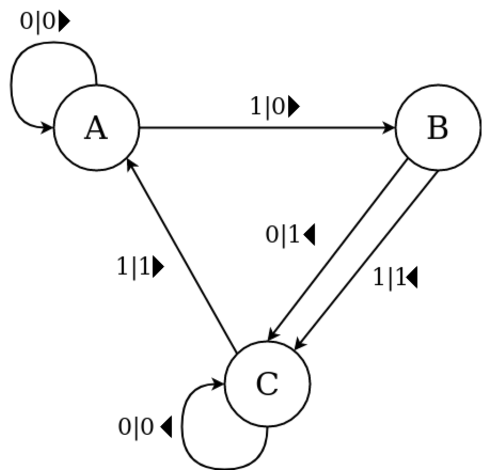

Figure 3.1: The Turing machine K_0. It uses the states A and C to search a 1 symbol in the tape

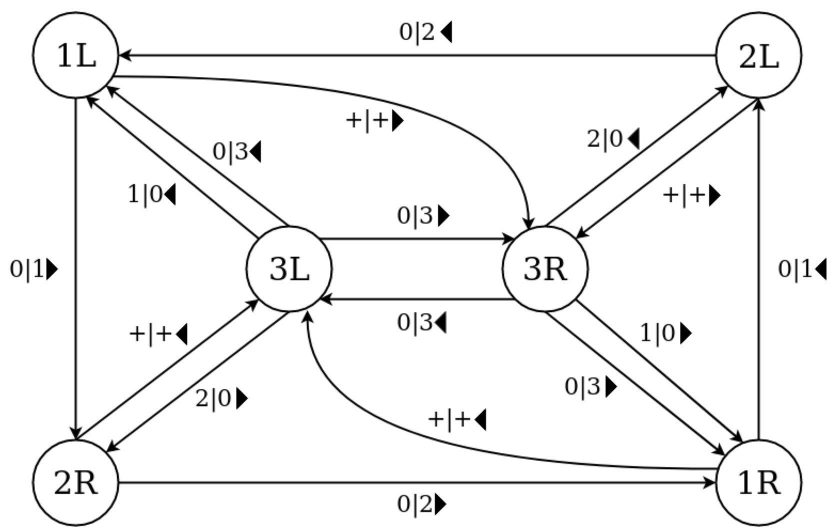

Figure 3.2: The Turing machine K_2. The symbol + represents any symbol of the alphabet of the machine except 0, thus $+|+\blacktriangleright$ is an abbreviation for $1|1\blacktriangleright$, $2|2\blacktriangleright$, $3|3\blacktriangleright$.

Now, let us explain how they proved the aperiodicity of the K_1 Turing machine. Keep in mind that this machine works with the alphabet $\Sigma = \{0, 1, 2, 3\}$ and its states are denoted by q_{ij} for $i \in \{1, 2, 3, 1', 2', 3'\}$ and $j \in \{1, 2, 3, 4, 5, 6\}$. The states q_{1j}, q_{2j}, and q_{3j} are space-symmetrical to $q_{1'j}, q_{2'j}$, and $q_{3'j}$, respectively. It means that they read and write the same symbols but they move in opposite directions.

First of all, they describe the machine's behavior.

Proposition 3.1 ([2])**.**

- *For any $s \geq 0$, let us define the following propositions:*
  $Q_1(s) : \left(\underset{q_{11}}{0}\ 0^s\ \alpha\ \beta \right) \vdash^* \left(0^{s+1}\ 0\ \underset{q_{16}}{\beta} \right)$
  $Q_2(s) : \left(\alpha\ 0^s\ \underset{q_{i1}}{0} \right) \vdash^* \left(\underset{q_{i6}}{\alpha}\ 0^{s+1} \right)$
  with $\alpha \in \{1, 2, 3\}$, $\beta \in \{0, 1, 2, 3\}$ and $(i = 2 \vee i = 3)$.

- *For any $k \geq 2$, for any integers $t, p, n \geq 0$ such that $t + p + n + 2 = k$ and for any $\alpha, \beta, \gamma \in \{1, 2, 3\}$, let us define the following proposition:*
  $P(k) : \left(\alpha\ 0^t\ \underset{q_{11}}{0}\ 0^p\ \gamma\ 0^n\ \beta \right) \vdash^* \left(\alpha\ 0^k\ \underset{q_{36}}{\beta} \right)$

The space-symmetric statements also holds for these propositions.

They use these propositions to prove the following results.

Lemma 3.1 ([2])**.** *The configurations $\left(\underset{q_{ij}}{0}\ 0^\omega \right)$, with $i \in \{1, 2', 3'\}$ and $j \in \{1, 2, 3\}$, are not periodic. This is also true for the space-symmetrical case.*

They prove it by showing that every configuration of that form will evolve into $\left(\alpha\ \underset{q_{11}}{0}\ 1\ 0^\omega \right)$, for some $\alpha \in \{1, 2, 3\}$. This configuration is of the type $\left(\alpha\ \underset{q_{11}}{0}\ 0^p\ \beta\ 0\ 0^\omega \right)$, for some $\alpha, \beta \in \{1, 2, 3\}$. Then, for any $p \geq 0$ we have:

$$\begin{aligned}
&\left(\alpha\ \underset{q_{11}}{0}\ 0^p\ \beta\ 0\ 0^\omega \right) \\
&\qquad\qquad \text{Apply } Q_1(p) \\
&\left(\alpha\ 0\ 0^p\ 0\ \underset{q_{16}}{0}\ 0^\omega \right) \\
&\qquad\qquad \text{One step} \\
&\left(\alpha\ 0\ 0^p\ \underset{q_{21}}{0}\ 1\ 0^\omega \right) \\
&\qquad\qquad \text{Apply } Q_2(p) \\
&\left(\underset{q_{26}}{\alpha}\ 0\ 0^p\ 0\ 1\ 0^\omega \right) \\
&\qquad\qquad \text{One step} \\
&\left(\alpha\ \underset{q_{11}}{0}\ 0^p\ 0\ 1\ 0^\omega \right)
\end{aligned}$$

We can iteratively apply this result and the number of 0s between α and the 1 symbol will increase, implying an aperiodic orbit.

Lemma 3.2 ([2])**.** *For any $n \geq 1$, let C_n be the set defined by $C_n = \{x | x \in \left(\underset{q_{i1}}{0}\ 0^{n-1} \right) | i \in \{1, 2', 3'\} \cup \left(0^{n-1}\ \underset{q_{i'1}}{0} \right) | i' \in \{1', 2, 3\}\}$. Starting with a configuration from C_n, the machine will eventually reach a configuration in C_{n+1}.*

They prove it combinatorially considering any possible scenario. Finally, they prove the main result.

Theorem 3.1 ([2])**.** *The machine K_1 has no periodic configuration.*

They demonstrate this proving that, starting from any configuration, the machine will eventually reach a configuration with a 0 as the read symbol and with an inner state of the form q_{i1} with $i \in \{1, 2, 3, 1', 2', 3'\}$. Then we can apply Lemma 3.2, so the machine can reach a configuration from C_n for any n. At some point, either the machine will reach a configuration of the form $\left(\begin{smallmatrix} 0 & 0^\omega \\ q_{i1} & \end{smallmatrix}\right)$ (or its symmetric), which by Lemma 3.1 is not periodic; or it will pass by an infinite sequence of configurations of the form $\left(\begin{smallmatrix} 0 & 0^{n-1} & \alpha \\ q_{i1} & & \end{smallmatrix}\right)$(or its symmetric), with $\alpha \neq 0$ and increasing n, implying a not periodic behavior.

However, the proposed machines did not have a property with particular importance in computation: reversibility.

3.3 The SMART Machine

In the article 'A small minimal aperiodic reversible Turing machine' [3] another example of an aperiodic Turing machine is shown, but in this case, the machine, that also works with bounded searches, results to be a reversible machine. This machine is nicknamed 'SMART' (Figure 3.3) and it has four states and works with three symbols. The authors go even further, proving that the SMART machine has more properties than aperiodicity and reversibility, it is also minimal (then transitive), time-symmetric, and it has a substitutive t-shift. Let us show their results.

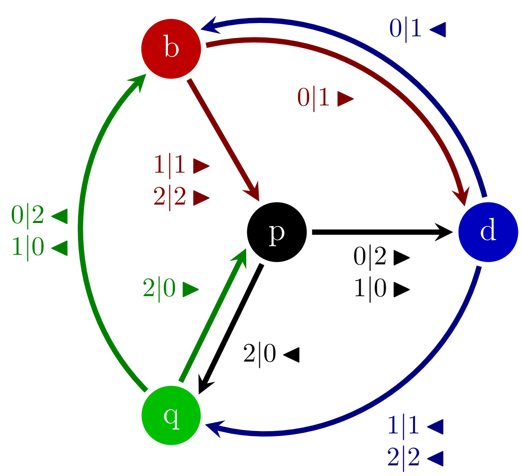

Figure 3.3: The SMART machine. An arrow from r to r' labelled $\alpha|\alpha' c$ represents the instruction $(r, \alpha, \alpha', r', c)$ of the machine.

3.3.1 Aperiodicity

First of all, the authors describe the general behavior of the machine proving the following propositions.

Proposition 3.2 ([3])**.**

$$
\begin{aligned}
&B(n) : (\forall s_+ \in \{1,2\})(\forall s \in \{0,1,2\}) \left(\begin{smallmatrix} s & 0^n & 0 & s_+ \\ & & b & \end{smallmatrix}\right) \vdash^* \begin{pmatrix} s & 0^{n+1} & s_+ \\ b & & \end{pmatrix} \\
&D(n) : (\forall s_+ \in \{1,2\})(\forall s \in \{0,1,2\}) \left(\begin{smallmatrix} s_+ & 0 & 0^n & s \\ & d & & \end{smallmatrix}\right) \vdash^* \begin{pmatrix} s_+ & 0^{n+1} & s \\ & & d \end{pmatrix} \\
&P(n) : (\forall s_+ \in \{1,2\}) \left(\begin{smallmatrix} 0 & 0^n & s_+ \\ p & & \end{smallmatrix}\right) \vdash^* \begin{pmatrix} 0^{n+1} & s_+ \\ & p \end{pmatrix} \\
&Q(n) : (\forall s_+ \in \{1,2\}) \left(\begin{smallmatrix} s_+ & 0^n & 0 \\ & & q \end{smallmatrix}\right) \vdash^* \begin{pmatrix} s_+ & 0^{n+1} \\ q & \end{pmatrix}
\end{aligned}
$$

They use this proposition to prove aperiodicity in two particular and important points.

Lemma 3.3 ([3])**.** $\left(\begin{smallmatrix} s_+ & 0 & 0^n & 1 & 0 \\ & p & & & \end{smallmatrix}\right) \vdash^* \begin{pmatrix} s_+ & 0 & 0^{n+1} & 1 \\ & p & & \end{pmatrix}$

This result implies the aperiodicity of the finite configuration $\left(\begin{smallmatrix} s_+ & 0 & 0^n & 1 & 0 \\ & p & & & \end{smallmatrix}\right)$ by using a similar reasoning to the one used in Lemma 3.1.

Lemma 3.4 ([3])**.** *The semi-infinite configurations* $\left(\begin{smallmatrix} 0 & 0^\omega \\ b & \end{smallmatrix}\right)$ *and* $\left(\begin{smallmatrix} 0^\omega & 0 \\ & d \end{smallmatrix}\right)$ *are not periodic.*

Proof. Starting with one of this configurations, after at most 9 steps, the machine will reach a configuration of the form $\left(\begin{smallmatrix} s_+ & 0 & 1 & 0^\omega \\ & p & & \end{smallmatrix}\right)$. Then we can iteratively apply Lemma 3.3 to see that the machine's evolution cannot be periodic.

■

Lemma 3.5 ([3])**.** *If we define, for every $n \geq 0$, the set $C_n = \{x | x \in \left(\begin{smallmatrix} s_+ & 0 & 0^n \\ & q & \end{smallmatrix}\right) \cup \left(\begin{smallmatrix} 0^n & 0 & s_+ \\ & p & \end{smallmatrix}\right)\}$, then for every $x \in C_n$, either x or the orbit of x will eventually visit C_m for arbitrary large m.*

This result basically means that every configuration belonging to C_n will increase the 0 symbols of the tape. To prove this, the authors use an combinatorial point of view and try every possible scenario, similar to Lemma 3.2.

Theorem 3.2 ([3])**.** *The SMART machine has no periodic points.*

The argument to prove this theorem is similar to Theorem 3.1. They prove that after less than 9 steps, the machine will be reading a 0 symbol in either state q or p. At this point, the current configuration can be manipulated in order to reach one of the sets C_n defined in Lemma 3.5. The amount of 0s will grow then, expanding to the left or to the right. At some point, the machine will either reach a configuration of the form $\left(\begin{smallmatrix} 0 & 0^\omega \\ r & \end{smallmatrix}\right)$, with $r \in \{b, p\}$ (or its symmetric), which is aperiodic by Lemma 3.4, or it will pass by an infinite sequence of configurations of the form $\left(\begin{smallmatrix} 0 & 0^{n-1} & s_+ \\ r & & \end{smallmatrix}\right)$, with $r \in \{b, p\}$ (or its symmetric), implying that its behavior is not periodic.

3.3.2 Other properties of the SMART machine

In addition to aperiodicity, the SMART machine has more interesting properties, these are time-symmetry, topological transitivity, topological minimality and substitutivity.

Proposition 3.3 ([3])**.** *The SMART machine is time-symmetric.*

Proof. Using involutions: $h_\Sigma(0) = 0$, $h_\Sigma(1) = 2$, $h_Q(d) = q$ and $h_Q(b) = p$, we will have that the SMART machine is time-symmetric. ∎

With that result, the authors prove the following lemma in order to demonstrate the topological minimality of the machine.

Lemma 3.6 ([3]). *For every finite word $v' \in \{0,1,2\}^*$ of length n, every $i \in \{1, ..., n\}$ and every $r \in Q$, there exists $k_1, k_2 \in \mathbb{N}$ such that* $\left(2\ 0^{n'}\ \underset{b}{0}\ 2\right) \vdash^* \left(2^{k_1}\ v'_1\ ...\ \underset{r}{v'_i}\ ...\ v'_n\ 2^{k_2}\right)$, *where* $n' = k_1 + k_2 + n - 3$.

We will not show the proof of this lemma here, but here is an important fact. Instead of directly prove this result, the authors prove that $\left(1^{k_1}\ h_\Sigma(v'_1)\ ...\ \underset{h_Q(r)}{h_\Sigma(v'_i)}\ ...\ h_\Sigma(v'_n)\ 1^{k_2}\right)$ reaches $\left(1\ 0^{n'}\ 0\ \underset{p}{1}\right)$, which, by time-symmetry, is equivalent to directly prove the lemma. In Chapter 4, we will see that the BinSmart machine is not time-symmetrical, so we cannot use this technique to prove any result.

3.3.3 An application of SMART

Furthermore, the authors show an application of the SMART machine demonstrating a conjecture from a previous work:

Conjecture 3.2 ([6]). *It is undecidable whether a given complete reversible Turing machine admits a periodic configuration.*

In order to prove this, they use two key techniques. The first one, *reversing the computation*, is taken from [6], and consist in a machine which in turn consists in a machine M together with its reverse machine, M^{-1}. The second technique is *embedding*, which consists in placing a machine inside a transition of another machine in such a way that the properties of the resulting machine depend on some properties of the original machines.

3.4 A New Aperiodic Machine

Afterward, a new Turing machine is proposed and it is nicknamed 'BinSmart' because it has similar behavior to the SMART machine but it works with a binary alphabet. This machine also uses bounded searches and it has and aperiodic behavior, but it is not a 'translation' of any other machine, so this makes a study necessary to know if the machine maintains the properties of its predecessor machine. We will study its properties in the following chapter.

Part III

Results

Chapter 4

The BinSmart Machine.

4.1 The BinSmart machine

In this section, we introduce the Binary Smart (BinSmart) machine which is the main object of study in this thesis. This Turing machine is based on another machine known as SMART machine [3], but it is not a 'translation' of it.

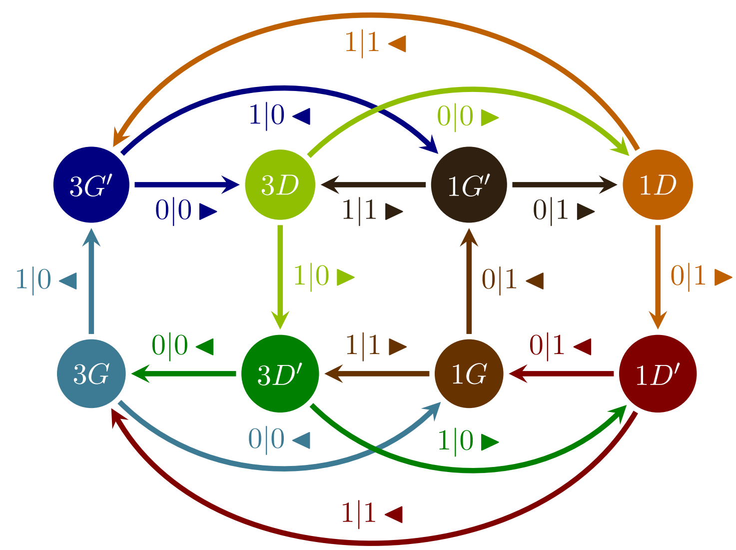

Figure 4.1: Binary Smart(BinSmart): We called it this way because it has a similar behavior than SMART machine, but with just two symbols. An arrow from r to r' labelled $\alpha|\alpha'c$ represents the instruction $(r, \alpha, \alpha', r', c)$ of the machine.

We remark the symmetry between states $1D$ and $1G$, between $1D'$ and $1G'$, between $3D$ and $3G$ and between $3D'$ and $3G'$. For example, the states $1D$ and $1G$ read and write exactly the same symbols but have opposite moving direction; this can be extended to the rest of the states.

4.2 BinSmart's behavior

The behavior of the BinSmart machine consists of applying bounded searches of 1s. To describe this behavior, considering $s \in \{0, 1\}$, the following propositions are defined:

$$\begin{array}{ll}
D_1(n) : \left(\underset{1D}{0}\ 0^n\ 1\right) \vdash^* \left(\underset{3G}{1}\ 0^n\ 1\right) & G_1(n) : \left(1\ 0^n\ \underset{1G}{0}\right) \vdash^* \left(1\ 0^n\ \underset{3D}{1}\right) \\
D_1'(n) : \left(1\ \underset{1D'}{0}\ 0^n\ s\right) \vdash^* \left(1\ 0\ 0^n\ \underset{1D'}{s}\right) & G_1'(n) : \left(s\ 0^n\ \underset{1G'}{0}\ 1\right) \vdash^* \left(\underset{1G'}{s}\ 0^n\ 0\ 1\right) \\
D_3(n) : \left(\underset{3D}{0}\ 0^n\ 1\right) \vdash^* \left(0\ 0^n\ \underset{3D}{1}\right) & G_3(n) : \left(1\ 0^n\ \underset{3G}{0}\right) \vdash^* \left(\underset{3G}{1}\ 0^n\ 0\right) \\
D_3'(n) : \left(1\ 0^n\ \underset{3D'}{0}\right) \vdash^* \left(\underset{3G}{1}\ 0^n\ 0\right) & G_3'(n) : \left(\underset{3G'}{0}\ 0^n\ 1\right) \vdash^* \left(0\ 0^n\ \underset{3D}{1}\right)
\end{array}$$

Lemma 4.1. *$D_1(n)$, $G_1(n)$, $D_1'(n)$, $G_1'(n)$, $D_3(n)$, $G_3(n)$, $D_3'(n)$ and $G_3'(n)$ are true for all $n \in \mathbb{N}$.*

Proof. Since $D_1(n)$, $D_1'(n)$, $D_3(n)$ and $D_3'(n)$ are symmetrical to $G_1(n)$, $G_1'(n)$, $G_3(n)$ and $G_3'(n)$, we will do the proofs just for the first ones. We prove $D_1'(n)$ and $D_3(n)$ by making an induction over n. The basis can be done by hand by simulating the machine. Let us suppose that these propositions are true for $n-1$ and for $n-2$. First we prove $D_1'(n)$.

$$\begin{array}{ll}
\left(1\ \underset{1D'}{0}\ 0^{n-3}\ 0\ 0\ 0\ s\right) & \\
 & \text{Apply } D_1'(n-1) \\
\left(1\ 0\ 0^{n-3}\ 0\ 0\ \underset{1D'}{0}\ s\right) & \\
 & \text{Two steps} \\
\left(1\ 0\ 0^{n-3}\ \underset{1G'}{0}\ 1\ 1\ s\right) & \\
 & \text{Apply } G_1'(n-2) \\
\left(\underset{1G'}{1}\ 0\ 0^{n-3}\ 0\ 1\ 1\ s\right) & \\
 & \text{One step} \\
\left(1\ \underset{3D}{0}\ 0^{n-3}\ 0\ 1\ 1\ s\right) & \\
 & \text{Apply } D_3(n-2) \\
\left(1\ 0\ 0^{n-3}\ 0\ \underset{3D}{1}\ 1\ s\right) & \\
 & \text{Two steps} \\
\left(1\ 0\ 0^{n-3}\ 0\ 0\ 0\ \underset{1D'}{s}\right) &
\end{array}$$

Now, for $D_3(n)$

$$
\begin{array}{ll}
\left(\underset{3D}{0}\ 0\ 0\ 0^{n-3}\ 0\ 1\right) & \\
 & \text{Two steps} \\
\left(0\ 1\ \underset{1D'}{0}\ 0^{n-3}\ 0\ 1\right) & \\
 & \text{Apply } D'_1(n-2) \\
\left(0\ 1\ 0\ 0^{n-3}\ 0\ \underset{1D'}{1}\right) & \\
 & \text{One Step} \\
\left(0\ 1\ 0\ 0^{n-3}\ \underset{3G}{0}\ 1\right) & \\
 & \text{Apply } G_3(n-2) \\
\left(0\ \underset{3G}{1}\ 0\ 0^{n-3}\ 0\ 1\right) & \\
 & \text{Two steps} \\
\left(0\ \underset{3D}{0}\ 0\ 0^{n-3}\ 0\ 1\right) & \\
 & \text{Apply } D_3(n-1) \\
\left(0\ 0\ 0\ 0^{n-3}\ 0\ \underset{3D}{1}\right) &
\end{array}
$$

Since $D_1(n)$ and $D'_3(n)$ are not recursive, we prove them directly. Let us prove $D_1(n)$.

$$
\begin{array}{ll}
\left(\underset{1D}{0}\ 0\ 0^{n-2}\ 0\ 1\right) & \\
 & \text{One Step} \\
\left(1\ \underset{1D'}{0}\ 0^{n-2}\ 0\ 1\right) & \\
 & \text{Apply } D'_1(n-1) \\
\left(1\ 0\ 0^{n-2}\ 0\ \underset{1D'}{1}\right) & \\
 & \text{One step} \\
\left(1\ 0\ 0^{n-2}\ \underset{3G}{0}\ 1\right) & \\
 & \text{Apply } G_3(n-1) \\
\left(\underset{3G}{1}\ 0\ 0^{n-2}\ 0\ 1\right) &
\end{array}
$$

Now, for $D'_3(n)$

$$
\begin{array}{ll}
\left(1\ 0^{n-1}\ 0\ \underset{3D'}{0}\right) & \\
 & \text{One Step} \\
\left(1\ 0^{n-1}\ \underset{3G}{0}\ 1\right) & \\
 & \text{Apply } G_3(n-1) \\
\left(\underset{3G}{1}\ 0^{n-1}\ 0\ 0\right) &
\end{array}
$$

■

4.3 Aperiodicity

Before proving that BinSmart does not have any periodic configuration, we prove aperiodicity in two particular but important points.

Lemma 4.2. $\left(1\ 0^n\ \underset{3D'}{1}\ 0\right) \vdash^* \left(1\ 0^n\ 0\ \underset{3D'}{1}\right)$

Proof.

$$\begin{array}{ll} \left(1\ 0^n\ \underset{3D'}{1}\ 0\right) & \\ & \text{Two Step} \\ \left(1\ 0^n\ \underset{1G}{0}\ 1\right) & \\ & \text{Apply } G_1(n) \\ \left(1\ 0^n\ \underset{3D}{1}\ 1\right) & \\ & \text{One step} \\ \left(1\ 0^n\ 0\ \underset{3D'}{1}\right) & \end{array}$$

■

Lemma 4.3. *The semi-infinite configuration* $\left(\underset{1D}{0}\ 0^\omega\right)$ *is not periodic.*

Proof. Starting with this configuration, the machine will evolve into $\left(1\ 0\ \underset{3D'}{1}\ 0^\omega\right)$ after 8 steps. Now we can apply Lemma 4.2 and see that the evolution of this configuration is in fact not periodic. ■

In order to generalize aperiodicity to any configuration, we will prove that arbitrary large blocks of 0s appear regardless of the context and in a recurrent way.

Lemma 4.4. *If we define, for every* $n \geq 0$*, the set* $C_n = \{x | x \in \left(\underset{1D}{0}\ 0^n\right) \cup \left(0^n\ \underset{1G}{0}\right)\}$*, then for every* $x \in C_n$*, either* x *or its orbit will eventually visit* C_m *for arbitrary large* m*.*

Proof. Since the states $1D$ and $1G$ are symmetrical, we just make the proof for the first one. We use $s_0, s_1, s_2, s_3 \in \{0, 1\}$ as variables.

$$\begin{array}{ll} \left(s_0\ s_1\ \underset{1D}{0}\ 0^n\ 1\ s_2\ s_3\right) & \\ & \text{Apply} D_1(n) \\ \left(s_0\ s_1\ \underset{3G}{1}\ 0^n\ 1\ s_2\ s_3\right) & \\ & \text{One step} \\ \left(s_0\ \underset{3G'}{s_1}\ 0\ 0^n\ 1\ s_2\ s_3\right) & \end{array}$$

if $s_1 = 0$

$$\begin{array}{ll} \left(s_0\ \underset{3G'}{0}\ 0\ 0^n\ 1\ s_2\ s_3\right) & \\ & \text{Apply } G_3'(n+1) \\ \left(s_0\ 0\ 0\ 0^n\ \underset{3D}{1}\ s_2\ s_3\right) & \\ & \text{One step} \\ \left(s_0\ 0\ 0\ 0^n\ 0\ \underset{3D'}{s_2}\ s_3\right) & \end{array}$$

if $s_2 = 1$

$$\left(s_0 \ 0 \ 0 \ 0^n \ 0 \ \underset{3D'}{1} \ s_3 \right)$$

One step

$$\left(s_0 \ 0 \ 0 \ 0^n \ 0 \ 0 \ \underset{1D'}{s_3} \right)$$

if $s_3 = 0$

$$\left(s_0 \ 0 \ 0 \ 0^n \ 0 \ 0 \ \underset{1D'}{0} \right)$$

One step

$$\left(s_0 \ 0 \ 0 \ 0^n \ 0 \ \underset{1G}{0} \ 1 \right)$$

We are done

if $s_3 = 1$

$$\left(s_0 \ 0 \ 0 \ 0^n \ 0 \ 0 \ \underset{1D'}{1} \right)$$

Two steps

$$\left(s_0 \ 0 \ 0 \ 0^n \ \underset{1G}{0} \ 0 \ 1 \right)$$

We are done

if $s_2 = 0$

$$\left(s_0 \ 0 \ 0 \ 0^n \ 0 \ \underset{3D'}{0} \ s_3 \right)$$

Two steps

$$\left(s_0 \ 0 \ 0 \ 0^{n-1} \ \underset{1G}{0} \ 0 \ 0 \ s_3 \right)$$

We are done

Now we study the case $s_1 = 1$

$$\left(s_0 \ \underset{3G'}{1} \ 0 \ 0^n \ 1 \ s_2 \ s_3 \right)$$

One step

$$\left(\underset{1G'}{s_0} \ 0 \ 0 \ 0^n \ 1 \ s_2 \ s_3 \right)$$

if $s_0 = 0$

$$\left(\underset{1G'}{0} \ 0 \ 0 \ 0^n \ 1 \ s_2 \ s_3 \right)$$

One step

$$\left(1 \ \underset{1D}{0} \ 0 \ 0^n \ 1 \ s_2 \ s_3 \right)$$

We are done

if $s_0 = 1$

$$\left(\underset{1G'}{1} \ 0 \ 0 \ 0^n \ 1 \ s_2 \ s_3 \right)$$

One step

$$\left(1 \ \underset{3D}{0} \ 0 \ 0^n \ 1 \ s_2 \ s_3 \right)$$

Apply $D_3(n+1)$

$$\left(1 \ 0 \ 0 \ 0^n \ \underset{3D}{1} \ s_2 \ s_3 \right)$$

One step

$$\left(1 \ 0 \ 0 \ 0^n \ 0 \ \underset{3D'}{s_2} \ s_3 \right)$$

if $s_2 = 1$

$\begin{pmatrix} 1\ 0\ 0\ 0^n\ 0 & \underset{3D'}{1} & s_3 \end{pmatrix}$

One step

$\begin{pmatrix} 1\ 0\ 0\ 0^n\ 0\ 0 & \underset{1D'}{s_3} \end{pmatrix}$

if $s_3 = 1$

$\begin{pmatrix} 1\ 0\ 0\ 0^n\ 0\ 0 & \underset{1D'}{1} \end{pmatrix}$

Two steps

$\begin{pmatrix} 1\ 0\ 0\ 0^n & \underset{1G}{0} & 0\ 1 \end{pmatrix}$

We are done

if $s_3 = 0$

$\begin{pmatrix} 1\ 0\ 0\ 0^n\ 0\ 0 & \underset{1D'}{0} \end{pmatrix}$

One step

$\begin{pmatrix} 1\ 0\ 0\ 0^n\ 0 & \underset{1G}{0} & 1 \end{pmatrix}$

We are done

if $s_2 = 0$

$\begin{pmatrix} 1\ 0\ 0\ 0^n\ 0 & \underset{3D'}{0} & s_3 \end{pmatrix}$

Two steps

$\begin{pmatrix} 1\ 0\ 0\ 0^{n-1} & \underset{1G}{0} & 0\ 0\ s_3 \end{pmatrix}$

We are done

■

Theorem 4.1. *The BinSmart machine has no periodic points.*

Proof. Consider an arbitrary configuration, after at most 11 steps, the machine will be reading a 0 symbol in either state $1D$ or $1G$, in other words, it arrives to one of the sets C_n defined in Lemma 4.4. The amount of 0s will grow then, expanding to the left or to the right. At some point the machine will either reach a configuration of the form $\begin{pmatrix} \underset{1D}{0} & 0^\omega \end{pmatrix}$ (or its symmetric), which we know to be aperiodic from Lemma 4.3, or it will pass by an infinite sequence of configurations of the form $\begin{pmatrix} \underset{1D}{0} & 0^n \end{pmatrix}$, with $n \geq 0$, (or its symmetric), implying that its behavior is not periodic. ■

4.4 Topological transitivity and minimality

We directly prove topological minimality which implies topological transitivity. To prove BinSmart minimality, we need to prove that every bi-infinite configuration reaches any finite configuration. To do that, we demonstrate that an arbitrary bi-infinite configuration x and an arbitrary finite configuration u reach another but 'identical' configuration v in different times. We also prove that u evolves into v faster than x. Since the machine is reversible, there is only one path to reach a specific configuration, then if x and u evolve into v, and u do it faster than x, implies that x reaches u. Since x and u are arbitrary, this argument is enough to prove minimality. To demonstrate this, the following lemmas are proved.

Lemma 4.5. *Every $x \in X$ will reach $\begin{pmatrix} \underset{1D}{0} & 0^p \end{pmatrix}$, for all $p \in \mathbb{N}$.*

Proof. As we know from Theorem 4.1, after at most 11 steps, any configuration arrives in one of the sets C_n for some n, then we apply Lemma 4.4 in order to reach C_m for any m, getting one of the following configurations (disregarding the context):

(i) $\begin{pmatrix} \underset{1D}{0} & 0^m \end{pmatrix}$

(ii) $\begin{pmatrix} 0^m & \underset{1G}{0} \end{pmatrix}$

If we reach (i), we are done, so let us see the case (ii):

$$\begin{array}{ll}
\left(0^m \; \underset{1G}{0}\right) & \\
& \text{Three Step} \\
\left(0^{m-1} \; \underset{3G'}{1} \; 1\right) & \\
& \text{Apply Lemma 4.2 } (m-2) \text{ times} \\
\left(0 \; \underset{3G'}{1} \; 0^{m-2} \; 1\right) & \\
& \text{Two steps} \\
\left(1 \; \underset{1D}{0} \; 0^{m-2} \; 1\right) & \\
& \text{We are done}
\end{array}$$

In this case we got that $p = m - 2$. Since we can reach this configuration for any m, we can do it for any p.

∎

Lemma 4.6. $\left(\underset{1D}{0} \; 0^n\right) \vdash^* \left(1 \; 0^m \; \underset{3D'}{1} \; 0^{n-m-1}\right)$, *for any* $m < n$

Proof. Starting from $\left(\underset{1D}{0} \; 0^n\right)$, after 3 steps we reach $\left(1 \; \underset{3D'}{1} \; 0^{n-1}\right)$, then we apply Lemma 4.2 m times, in an iterative way, to obtain: $\left(1 \; 0^m \; \underset{3D'}{1} \; 0^{n-m-1}\right)$. ∎

Lemma 4.7. *Considering an arbitrary finite word* $w \in \{0,1\}^*$ *of length* l*, the following statements are true:*

(i) $\left(1 \; 1 \; \underset{1D}{0} \; 0^n \; w_1 \; ... \; w_l \; 1\right) \vdash^* \left(1 \; 0 \; 0 \; 0^n \; 0 \; \underset{3D'}{w_2} \; ... \; w_l \; 1\right)$

(ii) $\left(1 \; w_1 \; ... \; w_l \; \underset{1D}{0} \; 0^n \; 1\right) \vdash^* \left(1 \; w_1 \; ... \; \underset{3G'}{w_l} \; 0 \; 0^n \; 1\right)$

(iii) $\left(1 \; w_1 \; ... \; w_l \; 0^n \; \underset{1G}{0} \; 1 \; 1\right) \vdash^* \left(1 \; w_1 \; ... \; \underset{3G'}{w_{l-1}} \; 0 \; 0^n \; 0 \; 0 \; 1\right)$

(iv) $\left(1 \; 0^n \; \underset{1G}{0} \; w_1 \; ... \; w_l \; 1\right) \vdash^* \left(1 \; 0^n \; 0 \; \underset{3D'}{w_1} \; ... \; w_l \; 1\right)$

Proof. Since (i) and (ii) are symmetrical to (iii) and (iv) we will do the proofs only for the first two.
Case (i):

$$\begin{array}{ll}
\left(1 \; 1 \; \underset{1D}{0} \; 0^n \; w_1 \; ... \; w_l \; 1\right) & \\
& \text{Apply } D_1(n) \; (w_1 \text{ have to be equal to } 1) \\
\left(1 \; 1 \; \underset{3G}{1} \; 0^n \; 1 \; w_2 \; ... \; w_l \; 1\right) & \\
& \text{3 steps} \\
\left(1 \; \underset{3D}{0} \; 0 \; 0^n \; 1 \; w_2 \; ... \; w_l \; 1\right) & \\
& \text{Apply } D_3(n+1) \\
\left(1 \; 0 \; 0 \; 0^n \; \underset{3D}{1} \; w_2 \; ... \; w_l \; 1\right) & \\
& \text{One step} \\
\left(1 \; 0 \; 0 \; 0^n \; 0 \; \underset{3D'}{w_2} \; ... \; w_l \; 1\right) & \\
& \text{We are done}
\end{array}$$

Case (ii):

$$\left(1\ w_1 \dots w_l\ \underset{1D}{0}\ 0^n\ 1 \right)$$

Apply $D_1(n)$

$$\left(1\ w_1 \dots w_l\ \underset{3G}{1}\ 0^n\ 1 \right)$$

One step

$$\left(1\ w_1 \dots \underset{3G'}{w_l}\ 0\ 0^n\ 1 \right)$$

We are done

■

Lemma 4.8. *Considering an arbitrary finite word $w \in \{0,1\}^*$ of length l, we got that:* $\left(1\ 1\ 1\ 0^n\ \underset{3D'}{w_1} \dots w_l \right) \vdash^* \left(1\ 0\ 0\ 0^{n+c+2}\ \underset{3D'}{w_{c+3}} \dots w_l \right)$*, where c is the amount of 0 symbols between w_1 and the first 1 symbol to the right.*

Proof. First, let us see the case $w_1 = 0$

$$\left(1\ 1\ 1\ 0^n\ \underset{3D'}{0}\ w_2 \dots w_l \right)$$

Apply $D_3'(n)$ $\qquad (*)$

$$\left(1\ 1\ \underset{3G}{1}\ 0^n\ 0\ w_2 \dots w_l \right)$$

Three steps

$$\left(1\ \underset{3D}{0}\ 0\ 0^n\ 0\ w_2 \dots w_l \right)$$

Apply $D_3(n+c+2)$

$$\left(1\ 0\ 0\ 0^n\ 0\ 0^c\ \underset{3D}{1}\ w_{c+3} \dots w_l \right)$$

One step

$$\left(1\ 0\ 0\ 0^n\ 0\ 0^c\ 0\ \underset{3D'}{w_{c+3}} \dots w_l \right)$$

We are done

Now, let us see the case $w_1 = 1$

$$\left(1\ 1\ 1\ 0^n\ \underset{3D'}{1}\ w_2 \dots w_l \right)$$

One step

$$\left(1\ 1\ 1\ 0^n\ 0\ \underset{1D'}{w_2} \dots w_l \right)$$

If $w_2 = 1$

$$\left(1\ 1\ 1\ 0^n\ 0\ \underset{1D'}{1}\ w_3 \dots w_l \right)$$

One step

$$\left(1\ 1\ 1\ 0^n\ \underset{3G}{0}\ 1\ w_3 \dots w_l \right)$$

Apply $G_3(n)$

$$\left(1\ 1\ \underset{3G}{1}\ 0^n\ 0\ 1\ w_3 \dots w_l \right)$$

Which reduces to $(*)$

If $w_2 = 0$, we can use Lemma 4.2 c times, obtaining $\left(\begin{smallmatrix} 1 & 1 & 1 & 0^{n+c} & 1 & 1 & w_{c+3} & \dots & w_l \\ & & & & 3D' & & & & \end{smallmatrix} \right)$, which reduces to the case $w_1 = w_2 = 1$. ∎

Corollary 4.1. $\left(\begin{smallmatrix} 1^k & 1 & 0^n & w_1 & \dots & w_l & 1 \\ & & & 3D' & & & \end{smallmatrix} \right) \vdash^* \left(\begin{smallmatrix} 1 & 0^{k+n+l} & 1 \\ & & 3D' \end{smallmatrix} \right)$*, with* $k = 2 \cdot |w|_1$.

Lemma 4.9. *Considering an arbitrary finite word* $w \in \{0,1\}^*$ *of length* m*, for every* $i \in \{1, 2, 3, ..., m\}$ *and every* $r \in Q$*, there exists* $k_1, k_2 \in \mathbb{N}$ *such that, every configuration of the form* $x = \left(\begin{smallmatrix} 1^{k_1+1} & w_1 & \dots w_i \dots & w_m & 1^{k_2+1} \\ & & r & & \end{smallmatrix} \right)$ *evolves into* $\left(\begin{smallmatrix} 1 & 0^{k_1+k_2+m} & 1 \\ & & 3D' \end{smallmatrix} \right)$ *or into* $\left(\begin{smallmatrix} 1 & 0^{k_1+k_2+m} & 1 \\ 3G' & & \end{smallmatrix} \right)$.

Proof. As we know from Theorem 4.1, every configuration evolves into a configuration that belongs to one of the sets C_n defined in Lemma 4.4, then the number of 0's will increase either to the right or to the left. We will call $w \in \{0,1\}^*$ to the part of the symbols that have not been turned into 0's, and l to the length of w. Then, we will reach one of the following configurations:

(i) $\left(\begin{smallmatrix} 1^{k_1+1} & 0 & 0^n & w_1 & \dots & w_l & 1^{k_2+1} \\ & 1D & & & & & \end{smallmatrix} \right)$

(ii) $\left(\begin{smallmatrix} 1^{k_1+1} & w_1 & \dots & w_l & 0 & 0^n & 1^{k_2+1} \\ & & & & 1D & & \end{smallmatrix} \right)$

(iii) $\left(\begin{smallmatrix} 1^{k_1+1} & w_1 & \dots & w_l & 0^n & 0 & 1^{k_2+1} \\ & & & & & 1G & \end{smallmatrix} \right)$

(iv) $\left(\begin{smallmatrix} 1^{k_1+1} & 0^n & 0 & w_1 & \dots & w_l & 1^{k_2+1} \\ & & 1G & & & & \end{smallmatrix} \right)$

where $n + l + 1 = m$. As before, we will do the proof only for the cases (i) and (ii) since they are symmetric to (iii) and (iv). At this point, we can apply Lemma 4.7 obtaining the following configurations:

(a) $\left(\begin{smallmatrix} 1^{k_1} & 0 & 0 & 0^n & 0 & w_2 & \dots & w_l & 1^{k_2+1} \\ & & & & & 3D' & & & \end{smallmatrix} \right)$ for (i)

(b) $\left(\begin{smallmatrix} 1^{k_1+1} & w_1 & \dots & w_l & 0 & 0^n & 1^{k_2+1} \\ & & & 3G' & & & \end{smallmatrix} \right)$ for (ii)

Now we apply Corollary 4.1 in order to reach the next configurations:

- $\left(\begin{smallmatrix} 1 & 0^{k_1+k_2+m} & 1 \\ & & 3D' \end{smallmatrix} \right)$ for (a)
- $\left(\begin{smallmatrix} 1 & 0^{k_1+k_2+m} & 1 \\ 3G' & & \end{smallmatrix} \right)$ for (b)

∎

Lemma 4.10. $\left(\begin{smallmatrix} 1 & 1 & 0^n & 1 & 1 \\ & 3G' & & & \end{smallmatrix} \right) \vdash^* \left(\begin{smallmatrix} 1 & 0^{n+2} & 1 \\ & & 3D' \end{smallmatrix} \right)$.

Proof.

$$\begin{array}{ll} \left(\begin{smallmatrix} 1 & \underset{3G'}{1} & 0^n & 1 & 1 \end{smallmatrix}\right) & \\ & \text{Two step} \\ \left(\begin{smallmatrix} 1 & \underset{3D}{0} & 0^n & 1 & 1 \end{smallmatrix}\right) & \\ & \text{Apply} D_3(n) \\ \left(\begin{smallmatrix} 1 & 0 & 0^n & \underset{3D}{1} & 1 \end{smallmatrix}\right) & \\ & \text{One step} \\ \left(\begin{smallmatrix} 1 & 0 & 0^n & 0 & \underset{3D'}{1} \end{smallmatrix}\right) & \\ & \text{We are done} \end{array}$$

■

Theorem 4.2. *The BinSmart machine is minimal.*

Proof. Let w be an arbitrary finite word of length l, $r \in Q$ an arbitrary state, $i \in \{1, 2, 3, ..., l\}$ and $x \in X$ an arbitrary bi-infinite configuration. It is enough to prove that the orbit of x contains the following configurations in the next order:

1. $t = \left(\begin{smallmatrix} \underset{1D}{0} & 0^n \end{smallmatrix}\right)$
2. $u = \left(\begin{smallmatrix} 1^{k_1+1} & w_1 & ...\underset{r}{w_i}... & w_l & 1^{k_2+1} \end{smallmatrix}\right)$
3. $v = \left(\begin{smallmatrix} 1 & 0^m & \underset{3D'}{1} & 0^{n-m-1} \end{smallmatrix}\right)$

for any $n, m, k_1, k_2 \in \mathbb{N}$, where $n > m$. Let us see the evolution.

- Evolution from x to t: done directly by Lemma 4.5.
- Evolution from t to v: done directly by Lemma 4.6.
- Evolution from u to v: using Lemma 4.9 and Lemma 4.10 for $l + k_1 + k_2 = m$
- Evolution from x to u: first of all, note that n is as big as we want, then always exists a path that is longer enough to include u before reach v, even if x contains v, all we have to do is to let it evolve into t with a big enough value of n. Considering this, that both x and u reach v, and the fact that the machine is reversible, so there is only one way to reach and specific configuration, we can deduce that x passes through u before evolving into v.

■

Corollary 4.2. *The BinSmart machine is transitive.*

4.5 Time-Symmetry

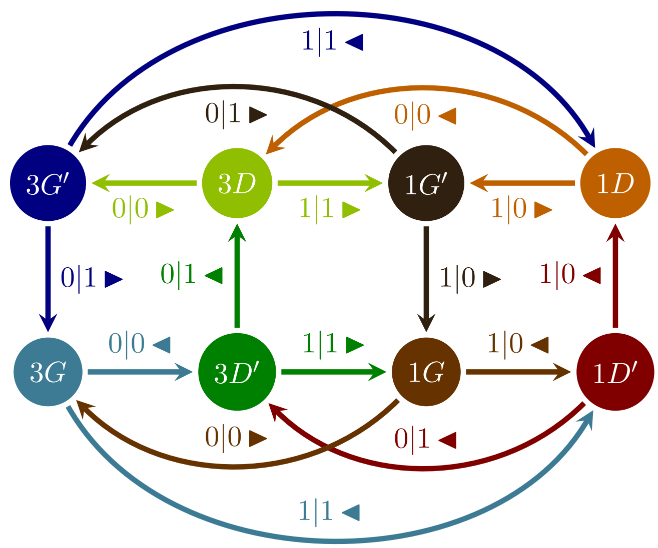

Figure 4.2: Reverse Binary Smart.

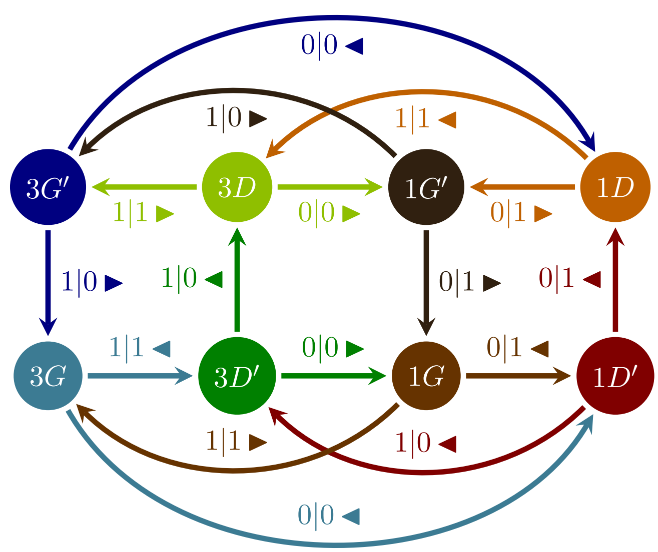

Figure 4.3: Reverse Binary Smart with involution $h_\Sigma : \{0 \to 1\}$

Proposition 4.1. *The BinSmart machine is not time-symmetric.*

Proof. We know that the instructions $(1D', 0, 1, 1G, \blacktriangleleft)$ and $(1D', 1, 1, 3G, \blacktriangleleft)$ are in δ, then, if the BinSmart machine is time-symmetric, there must exist two involutions h_Q and h_Σ such that $(h_Q(1D'), h_\Sigma(0), h_\Sigma(1), h_Q(1G), \blacktriangleleft)$ and $(h_Q(1D'), h_\Sigma(1), h_\Sigma(1), h_Q(3G), \blacktriangleleft)$ are in δ^{-1}. But there does not exist an involution h_Q that satisfies this condition neither with the involution $h_\Sigma : \{0 \to 0\}$ or with the involution $h_\Sigma : \{0 \to 1\}$. We can verify it with the help of the figures 4.1, 4.2 and 4.3. ■

Chapter 5

BinSmart's t-shift.

5.1 A substitutive t-shift

In this section, we prove that the BinSmart's t-shift is a substitutive subshift. First, we first need to define the following recursive functions.

- $S^1_D : \mathbb{N} \to (Q \times \Sigma)^*$
  $S^1_D(0) = \underset{1D'}{0}\,\underset{1G}{1}\,\underset{3D'}{1}$
  $S^1_D(1) = \underset{1D'}{0}\,\underset{1G}{1}\,\underset{3D'}{1}\,\underset{1D'}{0}\,\underset{1G}{0}\,\underset{1G'}{1}\,\underset{3D}{1}\,\underset{3D'}{1}$
  $S^1_D(n) = S^1_D(n-1)\,\underset{1D'}{0}\,\underset{1G}{0}\,S^1_G(n-2)\,\underset{1G'}{1}\,S^3_D(n-2)\,\underset{3D}{1}\,\underset{3D'}{1}$

- $S^1_G : \mathbb{N} \to (Q \times \Sigma)^*$
  $S^1_G(0) = \underset{1G'}{0}\,\underset{1D}{1}\,\underset{3G'}{1}$
  $S^1_G(1) = \underset{1G'}{0}\,\underset{1D}{1}\,\underset{3G'}{1}\,\underset{1G'}{0}\,\underset{1D}{0}\,\underset{1D'}{1}\,\underset{3G}{1}\,\underset{3G'}{1}$
  $S^1_G(n) = S^1_G(n-1)\,\underset{1G'}{0}\,\underset{1D}{0}\,S^1_D(n-2)\,\underset{1D'}{1}\,S^3_G(n-2)\,\underset{3G}{1}\,\underset{3G'}{1}$

- $S^3_D : \mathbb{N} \to (Q \times \Sigma)^*$
  $S^3_D(0) = \underset{3D}{0}\,\underset{1D}{1}\,\underset{3G'}{0}$
  $S^3_D(1) = \underset{3D}{0}\,\underset{1D}{0}\,\underset{1D'}{1}\,\underset{3G}{1}\,\underset{3G'}{0}\,\underset{3D}{0}\,\underset{1D}{1}\,\underset{3G'}{0}$
  $S^3_D(n) = \underset{3D}{0}\,\underset{1D}{0}\,S^1_D(n-2)\,\underset{1D'}{1}\,S^3_G(n-2)\,\underset{3G}{1}\,\underset{3G'}{0}\,S^3_D(n-1)$

- $S^3_G : \mathbb{N} \to (Q \times \Sigma)^*$
  $S^3_G(0) = \underset{3G}{0}\,\underset{1G}{1}\,\underset{3D'}{0}$
  $S^3_G(1) = \underset{3G}{0}\,\underset{1G}{0}\,\underset{1G'}{1}\,\underset{3D}{1}\,\underset{3D'}{0}\,\underset{3G}{0}\,\underset{1G}{1}\,\underset{3D'}{0}$
  $S^3_G(n) = \underset{3G}{0}\,\underset{1G}{0}\,S^1_G(n-2)\,\underset{1G'}{1}\,S^3_D(n-2)\,\underset{3D}{1}\,\underset{3D'}{0}\,S^3_G(n-1)$

Lemma 5.1. *$S^1_D(n)$ is the trace corresponding to applying the proposition $D'_1(n)$ to $\left(1\ \underset{1D'}{0}\ 0^n\ s\right)$ until $\left(1\ 0\ 0^n\ \underset{1D'}{s}\right)$. The analogous goes for $S^1_G(n)$.*

Similarly, $S^3_D(n)$ is the trace corresponding to applying the proposition $D_3(n)$ to $\left({0 \atop 3D} \; 0^n \; 1\right)$ until $\left(0 \; 0^n \; {1 \atop 3D}\right)$. The analogous goes for $S^3_G(n)$.

Proof. It is enough to see the proof of Lemma 4.1 and take the trace. ∎

Now let us define the substitution. Since the states $\{1D, 1D', 3D, 3D'\}$ are symmetrical to the states $\{1G, 1G', 3G, 3G'\}$, we only define the substitution for the first ones.
$\phi : (Q \times \Sigma)^* \to (Q \times \Sigma)^*$
$\phi({0 \atop 1D}) = {0 \atop 1D'}\,{1 \atop 1G}\,{1 \atop 3D'}$
$\phi({1 \atop 1D}) = {1 \atop 1D'}$
$\phi({0 \atop 1D'}) = {0 \atop 1D'}\,{0 \atop 1G}$
$\phi({1 \atop 1D'}) = {1 \atop 1D'}$
$\phi({0 \atop 3D}) = {0 \atop 3D}\,{0 \atop 1D}$
$\phi({1 \atop 3D}) = {0 \atop 3D}\,{1 \atop 1D}\,{0 \atop 3G'}$
$\phi({0 \atop 3D'}) = {1 \atop 3D}\,{0 \atop 3D'}$
$\phi({1 \atop 3D'}) = {1 \atop 3D}\,{1 \atop 3D'}$

For example, the substitution of ${0 \atop 1G'}\,{1 \atop 1D}\,{1 \atop 3G'}$ is $\phi({0 \atop 1G'}\,{1 \atop 1D}\,{1 \atop 3G'}) = {0 \atop 1G'}\,{0 \atop 1D}\,{1 \atop 1D'}\,{1 \atop 3G}\,{1 \atop 3G'}$.

Lemma 5.2. $S^1_D(n) = S^1_D(0)\phi(S^1_D(n-1))$
$S^1_G(n) = S^1_G(0)\phi(S^1_G(n-1))$
$S^3_D(n) = \phi(S^3_D(n-1))S^3_D(0)$
$S^3_G(n) = \phi(S^3_G(n-1))S^3_G(0)$

Proof. It is enough to prove it for $S^1_D(n)$ and $S^3_D(n)$, the other cases can be proved by symmetry.

$$\begin{aligned}
S^1_D(0)\phi(S^1_D(n)) &= S^1_D(0)\phi(S^1_D(n-1)\,{0 \atop 1D'}\,{0 \atop 1G}\,S^1_G(n-2)\,{1 \atop 1G'}\,S^3_D(n-2)\,{1 \atop 3D}\,{1 \atop 3D'}\,) \\
&= S^1_D(0)\phi(S^1_D(n-1))\,{0 \atop 1D'}\,{0 \atop 1G}\,{0 \atop 1G'}\,{1 \atop 1D}\,{1 \atop 3G'}\,\phi(S^1_G(n-2))\,{1 \atop 1G'}\,\phi(S^3_D(n-2))\,{0 \atop 3D}\,{1 \atop 1D}\,{0 \atop 3G'}\,{1 \atop 3D}\,{1 \atop 3D'} \\
&= S^1_D(0)\phi(S^1_D(n-1))\,{0 \atop 1D'}\,{0 \atop 1G}\,S^1_G(0)\phi(S^1_G(n-2))\,{1 \atop 1G'}\,\phi(S^3_D(n-2))S^3_D(0)\,{1 \atop 3D}\,{1 \atop 3D'} \\
&= S^1_D(n)\,{0 \atop 1D'}\,{0 \atop 1G}\,S^1_G(n-1)\,{1 \atop 1G'}\,S^3_D(n-1)\,{1 \atop 3D}\,{1 \atop 3D'} = S^1_D(n+1)
\end{aligned}$$

$$\begin{aligned}
\phi(S^3_D(n))S^3_D(0) &= \phi(\,{0 \atop 3D}\,{0 \atop 1D}\,S^1_D(n-2)\,{1 \atop 1D'}\,S^3_G(n-2)\,{1 \atop 3G}\,{0 \atop 3G'}\,S^3_D(n-1))S^3_D(0) \\
&= {0 \atop 3D}\,{0 \atop 1D}\,{0 \atop 1D'}\,{1 \atop 1G}\,{1 \atop 3D'}\,\phi(S^1_D(n-2))\,{1 \atop 1D'}\,\phi(S^3_G(n-2))\,{0 \atop 3G}\,{1 \atop 1G}\,{0 \atop 3D'}\,{1 \atop 3G}\,{0 \atop 3G'}\,\phi(S^3_D(n-1))S^3_D(0) \\
&= {0 \atop 3D}\,{0 \atop 1D}\,S^1_D(0)\phi(S^1_D(n-2))\,{1 \atop 1D'}\,\phi(S^3_G(n-2))S^3_G(0)\,{1 \atop 3G}\,{0 \atop 3G'}\,\phi(S^3_D(n-1))S^3_D(0) \\
&= {0 \atop 3D}\,{0 \atop 1D}\,S^1_D(n-1)\,{1 \atop 1D'}\,S^3_G(n-1)\,{1 \atop 3G}\,{0 \atop 3G'}\,S^3_D(n) = S^3_D(n+1)
\end{aligned}$$

∎

Theorem 5.1. *The t-shift of BinSmart is the closure of a fixed point of substitution ϕ.*

Proof. it is enough to prove that $\phi^n({0 \atop 1G'}) = {0 \atop 1G'}\,{0 \atop 1D}\,S^1_D(n-2)$ for all $n > 1$, because, from Lemma 5.1, ${0 \atop 1G'}\,{0 \atop 1D}\,S^1_D(n-2)$ is the trace of $\left({0 \atop 1G'}\;0^\omega\right)$ over the first steps and, as the configuration is transitive, the orbit of this configuration is dense. We will prove it by induction.

Base of induction: $\phi^2(\,{}^{0}_{1G'}\,) = {}^{0}_{1G'}\,{}^{0}_{1D}\,S^1_D(0)$
Induction hypothesis: $\phi^n(\,{}^{0}_{1G'}\,) = {}^{0}_{1G'}\,{}^{0}_{1D}\,S^1_D(n-2)$
Induction thesis:

$$\begin{aligned}
\phi^{n+1}(\,{}^{0}_{1G'}\,) &= \phi(\phi^n(\,{}^{0}_{1G'}\,)) \text{ // Induction hypothesis} \\
&= \phi(\,{}^{0}_{1G'}\,{}^{0}_{1D}\,S^1_D(n-2)) \\
&= {}^{0}_{1G'}\,{}^{0}_{1D}\,{}^{0}_{1D'}\,{}^{1}_{1G}\,{}^{1}_{3D'}\,\phi(S^1_D(n-2)) \\
&= {}^{0}_{1G'}\,{}^{0}_{1D}\,S^1_D(0)\phi(S^1_D(n-2)) \text{ // Lemma 5.2} \\
&= {}^{0}_{1G'}\,{}^{0}_{1D}\,S^1_D(n-1)
\end{aligned}$$

■

Chapter 6

Conclusion.

6.1 Conclusion

The BinSmart machine results to be another example of an aperiodic Turing machine in the TMT model, which means that once the machine evolves in a particular computation state, it will never be visited again. Moreover, the machine is also minimal, so 'starting' from any configuration, the machine has the ability to reach an arbitrary cylinder, which implies that the machine is also a transitive machine. Since the SMART machine is time-symmetrical, it is surprising that the BinSmart machine turns out to be not time-symmetrical; the last implies that it is distinguished when the machine evolves forward or backward in time. Finally, the t-shift associated with the BinSmart machine is a substitutive subshift, therefore we can define its dynamics in a different way. The Table 6.1 shows a summary of these results.

BinSmart Property	**Result**
Aperiodicity	✓
Topological Transitivity	✓
Topological Minimality	✓
Time-Symmetry	✗
Substitutive t-shift	✓

Table 6.1: Summary table of results.

As future work, we want to study the topological mixing of the BinSmart machine, that is a specific type of mixing. Mixing is an abstract concept originating from physics as an attempt of describing the irreversible thermodynamic process of mixing in the everyday world: mixing paint, mixing drinks, etc. Topological mixing is defined as:

$$\forall u, v \in {}^*\Sigma \times Q \times \Sigma^*, \exists N \in \mathbb{N}, \forall i > N : T^i(\mathcal{O}([u])) \cup [v] \neq \emptyset$$

It means that for any possible pair of cylinders, we can pick one of them so that after certain point, its image intersects the other for every iteration.

Bibliography

[1] Akin, E., Carlson, J.D.: Conceptions of topological transitivity. Topology and its Applications 159(12), 2815–2830 (2012)

[2] Blondel, V.D., Cassaigne, J., Nichitiu, C.: On the presence of periodic configurations in turing machines and in counter machines. Theoretical Computer Science 289(1), 573–590 (2002)

[3] Cassaigne, J., Ollinger, N., Torres-Avilés, R.: A small minimal aperiodic reversible turing machine. Journal of Computer and System Sciences 84, 288–301 (2017)

[4] Gajardo, A., Ollinger, N., Torres-Avilés, R.: The transitivity problem of turing machines. In: International Symposium on Mathematical Foundations of Computer Science. pp. 231–242. Springer (2015)

[5] Hooper, P.K.: The undecidability of the turing machine immortality problem 1. The Journal of Symbolic Logic 31(2), 219–234 (1966)

[6] Kari, J., Ollinger, N.: Periodicity and immortality in reversible computing. In: International Symposium on Mathematical Foundations of Computer Science. pp. 419–430. Springer (2008)

[7] Kůrka, P.: On topological dynamics of turing machines. Theoretical Computer Science 174(1-2), 203–216 (1997)

[8] Kůrka, P.: Topological and symbolic dynamics, vol. 11. SMF (2003)

[9] Lind, D., Marcus, B., Douglas, L., Brian, M.: An introduction to symbolic dynamics and coding. Cambridge university press (1995)

[10] Moore, C.: Unpredictability and undecidability in dynamical systems. Physical Review Letters 64(20), 2354 (1990)

[11] Torres, R., Ollinger, N., Gajardo, A.: Undecidability of the surjectivity of the subshift associated to a turing machine. In: International Workshop on Reversible Computation. pp. 44–56. Springer (2012)

[12] Turing, A.M.: On computable numbers, with an application to the entscheidungsproblem. Proceedings of the London mathematical society 2(1), 230–265 (1937)

[13] Viro, O.Y., Ivanov, O., Netsvetaev, N.Y., Kharlamov, V.: Elementary topology. American Mathematical Soc. (2008)