THÈSE

Pour obtenir le grade de

DOCTEUR DE l'UNIVERSITÉ AIX-MARSEILLE

Algorithmic complexity of growth-type invariants of SFT under dynamical constraints

and principles of information processing under these constraints

Spécialité **Mathématiques**

École doctorale de Mathématiques et Informatique

Par **Silvère Gangloff**, soutenue publiquement le **28 juin 2018**.

COMPOSITION DU JURY :

MATHIEU SABLIK	Université Paul Sabatier	Directeur
GUILLAUME THEYSSIER	CNRS et Aix-Marseille Université	Directeur
EMMANUEL JEANDEL	Université de Lorraine	Rapporteur
MICHAEL HOCHMAN	Hebrew university	Rapporteur
VALÉRIE BERTHÉ	CNRS et Université Paris Diderot	Examinatrice
JÉRÔME BUZZI	CNRS et Université Paris Sud	Examinateur
ANDREI ROMASHCHENKO	CNRS et Université de Montpellier	Examinateur
SAMUEL PETITE	Université de Picardie	Examinateur

Abstracts

Les sous-décalages multidimensionnels (SFT) sont des ensembles de colorations d'une grille régulière infinie ($\mathbb{Z}^d, d \geq 2$) définis par lois locales. Ces objets sont impliqués dans plusieurs domaines des mathématiques et en particulier la physique statistique. Un problème important dans ce cadre est de trouver une formule pour l'entropie. Excepté pour quelques modèles, une telle formule n'est pas connue. De plus, il a été démontré relativement récemment qu'il n'existe pas de méthode pour calculer (algorithmiquement) l'entropie de ces systèmes. Par conséquent, une approche naturelle au problème est de chercher une sous-classe (maximale) sur laquelle il existe un algorithme permettant de calculer uniformément l'entropie.

Dans ce texte, on propose des résultats reliés à ce problème. Les plus important d'entre eux sont: (1) L'existence de SFT apériodiques vérifiant une variante de la propriété d'assemblage introduite ici et appelée assemblage linéaire; (2) Une caractérisation des valeurs de l'entropie pour ces SFT (répondant en particulier à une question ouverte de Hochman et Meyerovitch); (3) Une caractérisation d'un seuil pour la calculabilité de l'entropie pour les sous-décalages multidimensionnels à language décidable; (4) Une caractérisation des valeurs possibles de la dimension entropique pour les SFT tridimensionnels minimaux.

Les deux résultats de caractérisation peuvent être interprétés comme une approximation par le haut de classes de SFT optimales sur lesquelles l'entropie (resp. la dimension entropique) peut être calculé uniformément, étendant le domaine dynamique sur lequel un calcul universel peut être implémenté dans les SFT. Le résultat sur le seuil caractérise une classe optimale mais pour les sous-décalage à language décidable, classe un peu plus souple que celle des SFT. Un problème important reste ouvert: celui de trouver une classe optimale pour les SFT multidimensionnels. De plus, on étend le problème à l'intervalle unité, et caractérise les valeurs possibles de l'entropie pour les fonctions calculables de l'intervalle.

Mots-clé: sous-décalages, entropie, invariants topologiques, calculabilité.

Multidimensional subshifts of finite type (SFT) are sets of colorings of an infinite regular grid ($\mathbb{Z}^d, d \geq 2$) defined by local rules. These objects are involved in many areas of mathematics and in particular statistical physics. One important problem in this setting is to compute the entropy with a formula. Unfortunately, it was proved, quite recently, that there can not be a general method to compute (algorithmically) the entropy of these systems. The subsequent approach to the problem is to seek for a subclass on which the entropy can be computed algorithmically. That is the main motivation to study the algorithmical complexity of SFT in subclasses defined by dynamical constraints.

In this text, we expose results related to this problem. The most saliant ones are: (1) The existence of aperiodic subshifts of finite type under linear version block gluing constraint, introduced here; (2) A characterization of the values of the entropy for these subshifts (this answers an open question by Hochman and Meyerovitch); (3) A characterization of a threshold on a quantified version of the irreducibility property for the computability of the entropy of decidable subshifts; (4) A characterization of the entropy dimensions of minimal tridimensional SFT.

The two characterization results can be interpreted as the approximation from above of some classes of SFT in which the possibility to embed universal computation (and thus have the non-computability of the entropy) vanishes. The result on the threshold characterize an optimal class but is related to decidable subshifts. A challenging open question is to find a similar result for multidimensional SFT. Moreover, we extend this investigation on the unit interval and characterize the values of the entropy of computable interval maps.

The introduction of this text and its conclusive part include an interpretation and analogies that relate the problem and objects considered here with biology, in the spirit of cybernetics.

Keywords subshifts, entropy, growth-type invariants, computability.

Acknowledgements

First of all I would like to express my sincere gratitude to my main advisor, Prof. Mathieu Sablik, mostly for his patience, for his availability and support during these three years, and moreover for what he taught me about how to write mathematics, for the excitement I see now in writing a mathematical text, which at some level, demands - and this is what I find exciting in writing - to understand the complex cognitive processes triggered while reading the text (in order to make the text clear for the reader one has to adapt to these processes). This had a significant impact on my way of teaching, and how I relate now the way I practice research and the way I practice teaching. I also thank Mathieu for teaching me how to concentrate on a precise problem, by constantly recalling me to think back about the questions yet not answered - I am convinced that any significant answer can come out only by the constant concentration on some concise area of thoughts, and that this is due to the natural inertia of these thoughts, which is the main obstacle one has to overcome in order to create.

To my other co-authors, Benjamin Hellouin and Cristobal Rojas, with whom I spent enjoyable and fruitful first months of my PhD in Santiago, Chile. Thanks to Benjamin for his demonstration of adaptation of the questioning attitude to the opening will of the mathematical objects - while thinking together on what became the fourth chapter of my thesis.

To Prof. Alejandro Maass for his welcoming and the Centro de Modelamiento Matematico, Universidad de Chile, as well as the CNRS for financial support during these three months. I thank also the other dynamicians Guillaume Theyssier, Vincent Delecroix, Michael Schraudner, and Pierre Guillon for interesting discussions, and patient explanations about some hard-to-understand recent constructions. Thanks also to Laurent Perrinet for interesting discussion about possible connection of my matter to neuroscience at the INT in Marseille.

Thanks also to Guillaume Theyssier and Cristobal for careful reading of the preparatory version of this text and for helpful advices on writing, in particular the introductive part. To Prof. Emmanuel Jeandel and Prof. Michael Hochman for their time and patience for the reviewing the final version of the text.

Merci à Jean-Christophe Feauveau pour son excellent cours de mathématiques, qui a été l'occasion pour moi de commencer à apprécier le travail intellectuel et la recherche, certes élémentaire mais stimulante, de manière autonome. I would also like to thank my english teacher, from whom I remember a simple but powerful image about the structure of thoughts of the translation process : in order to translate from a language into another, you will have first to abstract in your mind the idea from the text, and *then* express the idea using the other language - this reflects the fact that we usually learn a language as a coherent structure, not in relation with another one. Now I see the wide extent of this image, and how a powerful tool this is for me even as a guide in any attempt to communicate about mathematics.

Bien sûr, je voudrais remercier ceux avec qui j'ai partagé ces années de doctorat, mes amis, Agathe en particulier pour avoir accepté de relire le texte et Jean-Quoissandre pour son soutien. À mes parents qui m'ont soutenu tout au long de ces années, and to June for being here everyday and for the purpose of this thesis for her reading of the language :) And from beyond the grave, thanks to C.G Jung

for *Psychology and Alchemy*, A. Turing for the Computing machines, and M. Merleau-Ponty for the *Perception phenomenology.*

Contents

Chapter 1

Introduction

That which is below is like that which is above and that which is above is like that which is below to do the miracles of one only thing **- The Emerald tablet, translation by I. Newton.**

The Emerald tablet is a hermetic text whose origin is unknown. During medieval ages it was believed that this text contained the secrets of the philosopher stone. The terms 'below' and 'above' could refer to the direction that goes from the microcosm to the macrocosm, and come along with an idea of hierarchy. This hierarchy relies on the composition relation that exists between beings who lie in different scales (beings are composed of beings of smaller scales). The sentence written above tells that the microcosm and the macrocosm are analogous and that even if they lie in different scales, all beings behave in a similar way.

This hermetic axiom is reflected in a remarkable way by the theoretical constructions of B. Durand, A. Romaschenko and A. Shen. These constructions follow the principle of J.V. Neumann's self-reproducing theoretical machines. However, in these constructions, the machine reproduces itself through some scale dimension: at each scale, coded by an integer number, a computing machine works in order to construct a symbolic coding of the machine which lies at the scale immediately above while executing proper computations. The description of these constructions relies on an inductive process, relating the dynamics of the machine at any scale to the scale immediately below.

In this text, we propose constructions that are close to these ones except that there is no causal relation between the machines that lie in different scales and all the mechanisms are described relatively to scale zero. Although this makes the constructions more complex to formulate in a general setting, it enables us to adapt them to stronger constraints.

The objects considered in this text are symbolic dynamical systems called **subshifts of finite type**. In Section 1.2, we give evidence that this class of systems provides a natural context for understanding the consequences of dynamics, and furthermore geometry, on computation. We give an overview on recent developments in this direction. In Section 1.3, we explain our contributions to this study.

Historical relations between computation theories and biology had a significant impact on this field, through analogies with computing machines. However, recent paradigms suggest us to take more into account the dynamical constraints of a living system on how it processes information. The systems studied here provide us with a mathematical counterpart to these paradigms. We explain this in Section 1.1. We also address questions for which analogies with subshifts dynamical systems could provide some insight.

1.1 Information processing in biological systems

The concept of **information** has been a central tool for understanding biological systems. Here we think of an information, from the point of view of a living system, as a stimulus perceived by this living system from its evolution context. In the mathematical setting considered in this text, information is measured in bits, and represented by symbols in a finite set. In this setting, the system **processes information**, in the sense that it transports and transforms bits of information through various mechanical processes inside itself. This information processing results in an action by the system as a response to the external stimulus on its evolution context. Since the system is **constrained** by its context, its information processing mechanisms are also constrained. For instance, when a predator is present in the context, the processing of this information has to be **fast**: this is an example of constraint. This constraint implies in particular that this type of information is **treated in a specific way**, since the brain can not process all the information in a fast way. This is an example of effect of the constraint on information processing.

The more complex is the context, the more information processing involves complex mechanisms and organization amongst them.

1.1.1 Bits of information and computation

Considering information as a set of bits (of information) enables to quantify it. The mathematical **entropy** was constructed in *A mathematical theory of communication (1948)* by C. Shannon in order to measure the maximal quantity of bits of information that a stochastic source can emit per time unit, in order to compress the information transmitted. In this sense, it measures the capacity of the source. It was then extended to more general (topological) dynamical systems in [AKM65].

On the other hand, the **Kolmogorov complexity** was developed in order to measure the minimal quantity of information required to describe some mathematical object. More precisely, given a universal interpreter of programs, the Kolmogorov complexity of an object is the minimal length of a program that describes this object.

The notion of **computing machines**, constructed by A. Turing in *On computable numbers, with application to the Entscheidungsproblem (1936)*, also relies on the conception of information as a set of bits : these machines manipulate, by *finite means*, a sequences of symbols in a finite set. The aim of A. Turing was to solve the D. Hilbert's algorithmical decidability problem by means of a negative answer, exhibiting problems that are undecidable.

1.1.2 Cybernetics

In the context of **cybernetics**, some other models were developed, sharing with Turing's computing machines the ability to process information, interpreted as a set of bits, by *finite means*.

The aim of cybernetics was, by analogies between theoretical, artificial systems and biological ones, to understand the gap that there exists between them in terms of information processing. In particular, it led to the creation of theoretical systems that exhibit properties analogous to some properties observed in living systems. This approach is abstracted in *Cybernetics, or Control and Communication in the Animal and the Machine*, written by N. Wiener.

One can find well-known illustrations of this movement in the works of A. Turing, *The Chemical Basis of Morphogenesis (1956)*, and J.V. Neumann and A.W. Burks, *Theory of self-reproducing automata (1966)*. A. Turing proposed a model for morphogenesis - the natural process of patterns formation - describing the interactions of two molecules populations. For this purpose, he used the cybernetic notion of **feedback**: the consequences of the actions of some animal or entity affecting in return on the animal or entity itself. J.V. Neumann and A.W. Burks reproduced, using a discrete theoretical device introduced by J.V. Neumann called cellular automaton, a particular aspect of living systems (like cells for instance), namely **self-replication**, meaning the construction by the system of an identical copy of itself. Another notable application of cybernetics to biology were the analysis of the information processing into cells, by G. Gamow in *Information transfer in the living cell (1955).*

Further developments of cybernetics led to artificial intelligence (AI) which, using theoretical machines and artificial neural networks, inspired by real neural networks, aims to reproduce high-level characteristics and abilities of the human mind (for instance, learning).

1.1.3 Computationalism

Under this influence, the analogies were systematized, sometimes with loss of meaning, using theoretical objects outside of the context where they have actually some meaning. For instance, the computational theory of mind [Fod00] states that the mind is an information processing system and that thoughts are computations. Other works [BJ09] compared the computation principles of bacteria and the ones of Turing machines.

The objection to a systematic use of A. Turing's computing machines in order to understand information processing in the living is that it only appeared as a minimal formulation in order to prove undecidability results. The idea that these machines capture the notion of computation - meaning that any computational process made by humans can be also done by a Turing machine - appeared later, formulated explicitly by S. Kleene as the Church thesis, nowadays called Church-Turing thesis. Moreover, this model is highly abstracted from any material context: it tells what can be computed, but not how.

Despite this, the analogies between theoretical and biological systems provided powerful insights in biology. For instance, the notion of feedback is still used in order to described the living, and the development of the AI originates in the cybernetics movement. That is the reason why we believe that analogies can still provide insight in biology, giving however more attention to the meaning of the considered objects.

1.1.4 Embodied cognition

In the *embodied cognition* paradigm [RTV91], in order to understand the mind, one can not think of it as isolated from the other parts of the system, performing computations with the only motivation of precision in the representation of its objects. High-level cognition is highly contingent upon mechanisms of the whole organism, as well as the world perceived by an individual. For A. Damasio [Dam17], known for his work on emotions, the mind is submitted to homeostasis, meaning to the imperative of equilibrium and organism processes regulation. Emotional processes allow individuals to restrict the attention and high level cognition to deficiencies of the system when needed.

1.1.5 Mathematical counterpart to embodiement

The approach followed in this text is similar to the embodiement paradigm. We would like to understand how computing machines implemented in subshifts of finite type adapt to dynamical constraints. The computation model is not rigid and allows adaptations: various aspects of the machines can be modified. For instance, accelerated Turing machines can execute some pre-definite operations instantaneously [PS15]. The application of dynamical constraints leads to information processing structures that are natural in a sense that we discuss in Chapter 8.

We develop in the text an analogy between some of the structures observed in our constructions and some information processing structures observed in the living. We discuss this analogy in Chapter 8.

1.1.6 Questions related to this analogy

In this section, we address questions that come naturally to the mind in the context of the analogy between the approach of this text and the embodiement paradigm. The aim of this text in not to give an answer to these questions. However, we comment on them using the constructions of this text in Chapter 8. The motivation behind these comments is to show that this analogy could be developed further, and provide fruitful new interactions between symbolic dynamics and biology.

A lot of living beings (or parts of them) appear to treat the information by mean of **hierarchical structures** (as the visual cortex, see [Ser13]): what is the advantage of these structures in terms of information processing that makes them widely present in the living ? This question has been addressed for instance in [ACCR14]. What kind of constraints explain the emergence of these structures ? Let us point out that hierarchies are involved in multiple recent mathematical constructions of discrete dynamical systems in which computations are embedded [[HM10], [Mey11], [DRS12],[Moz89],[Rob71]]. However, this is not the only form of information processing in the living (immune system for instance). This implies a natural interrogation: is there a trade-off between information processing by mean of hierarchical structures and without these ? What kind of principles allow us to explain the use of one of the two modes of communication instead of the other (synaptic communication vs diffusive communication in the brain for instance [AFSM95]) ?

Other features are widely observed, like the **functional specialization** of information processing units. This refers to the division of a living system (or a part of it), into coherent sub-units which are attributed with some function (for instance storage [DNA for instance] of information, computation [cortex] and transmission of the information [diffusive, hierarchical], in order to achieve the global goal of being. One particular aspect of the functional specialization is **modularity**. This means that the functional parts can be replaced by another one, which ensures the same function but by other means. See [Kan10] for references on functional specialization. Can we explain what kind of conditions trigger the functional specialization, when modularity is needed, and how this specialization is constructed ?

There are actually attempts to develop tools in order to understand the complexity of dynamical systems intuitively understood the organization and division into coherent units, like the work of K. Petersen and B. Wilson [[PW15]], inspired by works in neuroscience. This approach is global and consists in analyzing dynamical systems using global invariants. We would like to have a more local approach in the analysis of systems.

Another observable feature is the division of storage mechanism into coding and non coding parts, like DNA, whose non-coding and coding parts are respectively called introns and exons. Is there a function attributed to non-coding parts and by what means is it executed ? Various information processing structures in the living exhibit a particular geometry, like the existence of a nucleus into eukaryotic cells, enclosing (some) coding information, or the convolutions of the brain. Can we explain the geometry of these structure by how it influences computation ?

1.2 Subshifts dynamical systems

The systematic study of subshifts dynamical systems appeared in the work of M. Morse and G.A. Hedlund [HM38]. Their aim was to understand some properties of these systems, called transitivity and recurrence. The field they opened is called **symbolic dynamics**, the title of their paper.

1.2.1 Subshifts

A $\mathbb{Z}^d$-subshift, where $d \geq 1$, on some alphabet $\mathcal{A}$ - a finite set of symbols - is a set of elements of $\mathcal{A}^{\mathbb{Z}^d}$ which is closed according to the infinite power of the discrete topology on A and stable under the shift action. This last point means that for any element of this set, when all the symbols are shifted in a same direction, the obtained configuration of symbols is still in this set.

Equivalently, a subshift is a set of symbols displays which don't exhibit finite patterns in a predefinite set. These forbidden patterns are usually thought as **local rules**. These rules generate some **(global) behaviors**. The set of rules can be thought as a sequence of instructions that we write to program a subshift.

1.2.2 Subshifts dynamical systems as models for real systems

Subshifts systems can be interpreted as the result of a series of transformations on the conception that we may have of a real system.

A (mathematical) **dynamical system** is an abstraction that represents an object moving in a space according to a set of laws. In order to capture the possibilities for the notions of object, movement and space, the formal definition of a dynamical system involves:

- a (topological) space X, defined as a set having a notion of proximity of its elements (through topology or metrics),
- a function $T : X \times \mathbb{R}_+ \to X$ which integrate the set of laws restricting the movement of an object in the space X: if at time $t = 0$, the object is in position x_0, then at time $t \geq 0$ this object will be in position $T(x_0, t)$.

One can consider dynamical systems for which the time is discrete. In this setting, a system is the gathering of a space X and a function f. The trajectories are given by a function $T : X \times \mathbb{N} \to X$ which to some point x_0 of the space and to a time n associates $T(x_0, n) = f(f(...f(x_0)))$, where function f is applied n times. Historically, H. Poincaré considered these systems as an extraction from systems on a surface as return function in a region of the surface (see for instance [Dev03] for a reference).

One can also consider discretised space: the idea is, for a discrete time system (X, T) and a set of areas covering the space, to understand the dynamics of the systems through this covering. This means considering the dynamics of the system knowing only a partial (finite) information about the location of the object in the space.

Subshifts are a natural result of this series of abstractions, since they are discrete time systems whose space can be covered with set of regions that don't intersect. This fact simplifies the study of their dynamics.

From the the idea of extracting discrete objects from real systems derives the idea of programming some of these objects. This shift to programming was historically done in a remarkable way in the study of negative curvature surfaces and their geodesics. In this context, the discretisation method dates back to J. Hadamard [Had98], who proposed to code the geodesics of a surface with the elements of the fundamental group. Latter, M. Morse [Mor21] programmed a sequence, now called the Thue-Morse sequence, in order to produce some discontinuous geodesics that verify the dynamical property of recurrence. In the same vein, R. Grigorchuk [Gri80] programmed an automaton group which enabled him to solve Burnside's problem on periodic groups. In the context of dynamical systems, **subshifts**, presented in the next section, provide an intuitive context for programming dynamical systems.

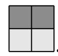
1.2.3 Programming subshifts

The meaning of the word *programming* is highly intuitive in the context of subshifts, for rules can code for some behaviors like loops (while and for), and branchings.

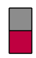
For instance, if the two symbols  and  are in the alphabet of the subshift (for instance thought of as representing the presence of some particle in some state), with the rule that the pattern

,

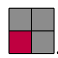
where light gray symbols stand for any symbol other than , implies the presence of the pattern

,

where the initial pattern appears on the left. Moreover, the pattern



implies the pattern

.

The global behavior induced by this rule is that the particle in dark gray state goes onto the right **while** there is not particle in purple state under. When this happens, the trajectory of the particle is shifted downwards. See Figure 1.1.

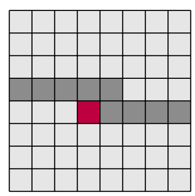


Figure 1.1: An exemple of global behavior induced by a while loop.

The instructions that we intuitively find in order to program a subshift are themselves often enumerated by an algorithm. In is case, the subshift is said to be **effective** . Although very intuitive to manipulate, these systems are also intuitively different from *natural* emergent systems: the rules are imposed in a teleological way, in order to induce some behavior thought before.

1.2.4 Subshifts of finite type

The stakes in the study of subshifts are to understand when the rules that we have in mind can be induced by a **finite set of rules**. When this is the case, the subshift is called **subshift of finite type (SFT)**.

This corresponds to the *finite means* of Turing's computing machines, and the intuitive notion of *autonomy* of a system. This means that instead of being programmed from the outside, these dynamical systems are programmed from the inside: the finite set of rules can be thought as resulting of interactions of elements of the system, represented by symbols in the alphabet of the subshift.

These objects are central in symbolic dynamics, whose main problem is to classify (hence understand), up to conjugacy, the subshifts of finite type. Subshifts of finite type appeared already in the work of C. Shannon, as models for discrete communication channels [Sha48], and in others applications to coding theory [Mar85], dynamics of geodesic flows and classification of the dynamics of Anosov and Axiom A diffeomorphisms [AW70],[Bow78], and partial answer of D. Rudolph to Furstenberg's conjecture on invariants measures for $\times 2$, $\times 3$ transformations of the Torus [Rud90].

This text is centered on the study of subshifts of finite type, and the subshifts we program here are presented by first listing the rules, then describing the global behavior induced, in a less formal way. The reader who want to have a quick look over these constructions should read the global behaviors sections, before eventually reading the local rules lists.

They are organized into *modules* that we call layers. This means that the alphabet of the subshift is some product of alphabets $\mathcal{A}_1 \times ... \times \mathcal{A}_k$. The layer corresponding to the alphabet $\mathcal{A}_i$ is the projection of the subshift over ${\mathcal{A}_i}^{\mathbb{Z}^d}$, each one programmed to have a particular function, and to interact with the other layers.

The global behavior often consists in multiple phenomena having various natures: signals, structures, computations, etc. These phenomena exhibit a strong analogy with elementary objects of cellular biology: could we distinguish in a formal way their intuitive natures ?

1.2.5 Embedding computations in subshifts of finite type

The *finite means* of Turing's computing machines correspond to local rules: for each of these machines, the evolution on a given input is determined by a finite set of rules, hence the space-time diagram of this machine is a subshift of finite type. However, in this model, computation is not autonomous: in order to execute the desired computations, the machine has to be initialized, and this can not be imposed immediately with simple local rules.

Numerous methods have been developed further in order to autonomize computation, making well initialized Turing computations emerge from local rules, in an uniform way, using for instance one-dimensional cellular automata, with additional properties, such as invertibility [Dub98]. In the context of subshifts of finite type, the constructions of R. Berger [Ber66], and then R. Robinson [Rob71] (a simplification of R. Berger's construction), developed historically to prove the undecidability results, allow us to program computations in an autonomous way, into hierarchical structures that appear systematically in some **aperiodic subshifts of finite type**.

We would like to note that E. Shrödinger, in his book *What is life?*, presented aperiodicity as one of the principles for storing genetic information, through an aperiodic crystal. Both J.D. Watson and F.D. Crick, who discovered the structure of DNA, acknowledged the book as a source of inspiration for their initial researches [Wat07]. However, it appears now that hierarchical structures are more used in the constructions than the aperiodicity. This leads to the question of the possibility of embedding computations without the use of hierarchical structures.

Then, the construction of R. Robinson was extensively used in order to explore the dynamical properties of the subshifts of finite type class of dynamical systems. A remarkable example is the characterization of the non-negative numbers that are the entropy of multidimensional subshifts of finite type: M. Hochman and T. Meyerovitch [HM10] proved that these numbers are exactly the non-negative Π_1-computable numbers, which are the numbers that can be approximated from above by an algorithm. This construction embeds Turing computations in order to control the frequency of some symbols generating the entropy. T. Meyerovitch [Mey11] characterized, using similar tools, the possible entropy dimensions of a multidimensional SFT: these are the Δ_2-computable non-negative numbers, that is to say the numbers that can be approximated by an algorithm without any control on the convergence speed. Various other works followed, like the characterization of the possible sets of periods of a multidimensional SFT [JV15].

B. Durand, A. Romaschenko, and A. Shen [DRS12], inspired by the work of P. Gacs [Gác01], developed a construction which provides, using the notion of autosimulation of SFT, a more general way to construct aperiodic subshifts in which it is possible to embed computation, from which Hochman and Meyerovitch's achievement derives. Other applications of this construction followed, as the work of L. Brown Westrick [BW17], who proved that any subshift defined as blocks of 1 symbols surrounded by 0 symbols, whose length is in a given effective subset of $\mathbb{N}$ is a sofic subshift, meaning a factor of a

SFT.

Let us note another approach to embed computations in dynamical systems which consists in embedding computations in continuous dynamical systems defined by differential equations. This approach has been developed by O. Bournez [BC08], [KM99], or [BGP13].

1.2.6 Computation under constraints

Since embedding computations in subshifts of finite type allows building highly organized behaviors, this context seems natural to study the effect of constraints on computation, and the adaptation of information processing structures to these constraints. In considering this question, we follow recent works exhibiting notable effects of topological and dynamical constraints on the possibilities for computation in subshifts of finite type.

1.2.6.1 Consequences of geometry on information processing

The effect of geometry, and in particular dimension, on the computability of the entropy, and as a consequence on the possible mechanisms that we can code in subshifts in order to process information, is clear for subshifts of finite type: the entropy $\mathbb{Z}$-SFT is uniformly computable, while this is not the case for $\mathbb{Z}^d$-SFT for $d \geq 2$ [HM10]. This comes from the fact that $\mathbb{Z}$-subshifts are described by finite automata [Fis75], while this is not true for multidimensional subshifts. Multidimension allows the existence of aperiodic SFTs in which one can embed universal computation. Moreover, it is known that the conjugacy of two sofic $\mathbb{N}$-subshifts is decidable [Wil73], while it is undecidable for multidimensional SFTs. This problem is still open for $\mathbb{Z}$-SFTs.

M. Hochman [Hoc09] proved that three dimensions is sufficient to simulate effective one-dimensional subshifts with subshifts of finite type. Formally, the effective $\mathbb{Z}$-subshifts are the projective subactions of a $\mathbb{Z}^3$ sofic subshifts. This result was improved later and independently by B. Durand, A. Romashchenko and A. Shen [DRS12], using the fixed-point paradigm, and N. Aubrun and M. Sablik [AS13], using R. Robinson's technique. They proved that two dimensions are sufficient. Although using different formulations, the principle of their construction is to embed computations of machines such that all the machines *cooperate*, and are organized into a command structure.

Another approach proposed by N. Aubrun and M.-P. Béal [AB12] in order to understand the effect of dimension consists in considering subshifts of finite type on intermediate (discrete) monoids, like infinite ranked trees. They proved that the conjugacy problem is decidable for these subshifts. This intuitively means this class is closer to the class of one-sided subshifts than to the class of two-sided ones.

Recent works proved the existence of aperiodic subshifts in various groups other than $\mathbb{Z}^d$, like the Baumslag-Solitar group [AK13]. It has been proved that a hyperbolic group admits a strongly aperiodic SFT if and only if it has one end [GS05]. These constructions still use hierarchical structures. This context seems to be adapted to study the influence of the group geometry on computation. Other authors extended M. Hochman's simulation theorem [Hoc09] for subshifts on groups defined as semi-products [BS16] or products of groups [Bar17], and as a particular case R. Grigorchuk's group. These constructions follow the general scheme of M. Hochman and T. Meyerovitch's construction, except that they use the presentation of the group to find a two-dimensional grid similar to $\mathbb{Z}^2$.

1.2.6.2 Consequences of dynamical constraints

The question of the effects of dynamical constraints comes from a question written in [HM10], where the authors asked if their result would be still true under the dynamical constraint of **transitivity**.

It has been studied in a work by R. Pavlov and M. Shraudner, who realized some sub-class of the computable numbers as entropy of three-dimensional SFT that verifies a mixing property which allows two patterns to appear in a configuration of the subshift which gap between them [PS15] greater than a constant gap. Their construction uses accelerated Turing machines, defined to have the possibility to execute a pre-defined set of operations in instant time, in order to produce a quasi-sturmian word on which are superimposed random bits generating the entropy. They use an operator that transforms subshifts into block gluing ones. The way they embed computation allows to have control over the complexity function of the subshift. With this control, they ensure that the entropy of the subshift is not changed after transformation. The sub-class that they realized corresponds to a condition on approximation properties. The obstruction comes from the structures restricting the time allowed to the machines for their computations. However, this is not a characterization, since the only known obstruction is a more flexible condition on the approximation properties. Hence the following question:

Question 1. *What are the possible entropies of block gluing two-dimensional subshifts of finite type ?*

The **minimality** constraint has been studied for $\mathbb{Z}^d$-subshifts (not necessarily subshifts of finite type). M. Hochman and P. Vanier [HV17] studied Turing degrees of minimal subshifts, proving that there exist minimal subshift whose spectrum consists of an uncountable number of cones with disjoint base. U. Jung, J. Lee and K.K. Park [JLKP17] provided a characterization of the numbers that are the entropy dimension of a $\mathbb{Z}^2$-subshift, adapting C. Grillenberger's construction [Gri73] to dimension two. Recently, B. Durand and A. Romashchenko [DR17b] proved that any minimal effective one-dimensional SFT is the projective sub-action of a minimal $\mathbb{Z}^2$-SFT. This construction, that still relies on the fixed-point paradigm, simulates every admissible pattern in all the configurations. This is allowed by the specific growth of the computing units, and by the functional specialization of these units.

Let us note that subshifts of finite type have been studied through other computational aspects, like the existence of recursive points [Mye74], degrees of computability [Sim14],[JV11] and other types of constraints such as the countability [ST13], which prevents from using Robinson SFT.

1.2.6.3 This text

The approach followed in this text is to study the computability of growth-type invariants, such as the *entropy* and *entropy dimension*, of subshifts of finite type under dynamical restrictions.

The interest of this question, relative to the embodiement paradigm presented earlier, lies in how this reveals natural structures and principles for the information processing under dynamical constraints, and how the Turing machine embedded into subshifts of finite type have to be adapted to these constraints. The application of constraints to computing machines has already been studied, for instance in [COTA17], where is exhibited a small aperiodic and minimal reversible Turing machine. However, this did not lead to highly organized behaviors: this machine is based on the research of symbols on a line.

1.3 Organization of the manuscript

Chapter 2 is an intuitive introduction to subshifts dynamical systems and their properties, and Chapter 3 exposes computability notions for subshifts and algorithmic properties of their growth type invariants. Most of the results and definitions presented in these chapters are known. However, we tried to make as explicit as possible implicit thought mechanisms involved in the manipulation of subshifts. In particular, in Chapter 3, we proposed a formalisation of some terms used to describe the constructions of subshifts.

We express there the problem of characterizing growth type invariants of subshifts of finite type under dynamical constraints in terms of computability properties.

In the other chapters we expose our original contributions, whose main theorems are stated in the next sections of this chapter.

1.3.1 Block gluing with gap function

1.3.1.1 Multidimensional SFT

We introduce in Chapter 4 a variation of the block gluing property of [PS15], adding a gap function. Roughly, for a function $f : \mathbb{N} \to \mathbb{N}$, a subshift is f-block gluing when two cuboid patterns of size n in the language of this subshift can be 'glued' whenever they are separated by at least $f(n)$. A subshift is linearly block gluing when it is f-block gluing for a function f such that there exists some $C > 0$ such that for all $n \geq 1$, $n/C \leq f(n) \leq Cn$.

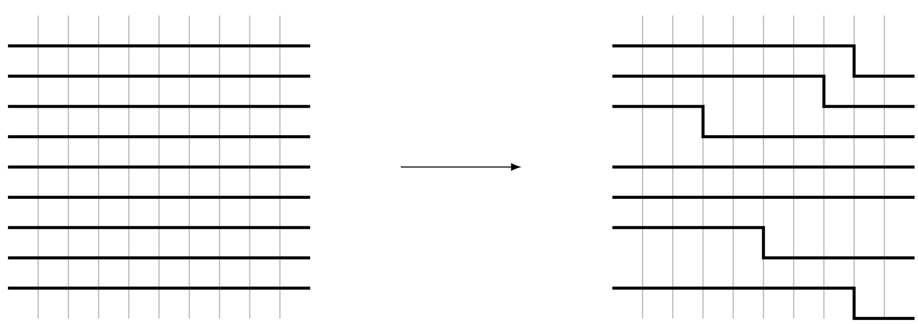
When f is constant, the f-block gluing property is the same as R. Pavlov and M. Shraudner's block gluing. This property implies that the entropy is computable.

We introduced this gap function in order to study the limit between the computable and the non-computable, and the effect of dynamical constraints on the non-computable. This problem is similar to some percolation problems, which consist in understanding the limit between two regimes for a parameter where a system has very different behaviors.

The main theorems of this chapter are the following ones. The first theorem provides the believed first ingredient towards autonomous computations in subshifts of finite type - aperiodicity, through hierarchical structures - under the constraint of linear block gluing. With the second one, we go further and state a characterization of the possible entropies under this constraint.

Theorem 1.1 ([GS17a]). *There exists an aperiodic linearly block gluing SFT.*

For the proof of this theorem, we introduce the notion of net gluing with gap function. This property means that two patterns can be glued one relatively to the other on a sub-lattice of $\mathbb{Z}^2$. We also introduce an operator on subshifts on some fixed alphabet. This operator transforms linearly net gluing subshifts into linearly block gluing ones. The principle of this transformation is to distort the graph of $\mathbb{Z}^2$, as illustrated on the following figure, multiple times and in different directions.



We then apply this transformation on the Robinson subshift which is known to be aperiodic and that we prove to be linearly net gluing.

Theorem 1.2 ([GS17a]). *The numbers that are the entropy of a linearly block gluing two-dimensional SFT are the Π_1-computable numbers.*

As linearly block gluing implies transitivity, this answers a question asked in [HM10], about the possibility to realize every Π_1-computable number as the entropy of a transitive SFT. This was also proved recently by B. Durand and A. Romaschenko [DR17a], using the fixed-point paradigm. This construction consists in coloring a part of every computing unit, with control on the frequency of the colored part in $\mathbb{Z}^2$. Random bits are superimposed to these colored parts, generating entropy. This subshift is also net gluing. However, its gap function f is superlinear: $f(n) = n^{\alpha}$, with $\alpha > 1$. The condition of linearity imposes restriction on the growth of the computing units. Since the fixed-point construction relies on this growth so that the computing machines have enough time to end their computations, the linearly net gluing property can not be derived directly from this paradigm.

The difficulty in the proof of this theorem comes from the fact that the construction by M. Hochman and T. Meyerovitch is very rigid:

1. the identification (which is necessary for the machines to work) of the symbols $0, 1$ is done over areas (infinite columns) unrelated to the structures which prevents the block gluing property: the obstruction is that a column of 1 symbols can not appear over a column of 0 symbols.
2. the machines implemented in the SFT can have degenerated (non-typical) behaviors, and this also prevents the gluing property.

In order to adapt the construction we need to change its general scheme. We need to adapt the identification mechanism to more local areas, and simulate the degenerated behaviors of the machines as well as normal (intended) behaviors. We allocate areas for normal machines and areas for simulating ones. A machine itself can not 'know' it has a simulation function, so we have to use error signals so that the machine exchanges information with another place in $\mathbb{Z}^d$ where this information is located. With these modifications, we construct linear net gluing SFT. In order to transform them into linearly block gluing ones, we adapt the transformation used to prove Theorem 1.1, so that the entropy of the image by this transformation can be expressed with the entropy of the initial subshift. A more detailed description of this proof can be found in Chapter 4, Section 4.4.3.

A notable aspect of this construction is that in order to keep the dynamical properties, we have to introduce parasitic entropy (which is not generated by random bits). However, this parasitic entropy can be imposed to be as small as we want, and the random bits serve to supplement this entropy into the aimed one.

The same phenomenon is observable while dealing with the dynamical constraint of minimality.

1.3.1.2 Decidable subshifts

In Chapter 5, we define a notion of irreduciblity with gap function. A subshift is said to be f-irreducible, for a function $f : \mathbb{N} \to \mathbb{N}$, when any two patterns - not necessarily cuboid ones - can be glued whenever the distance between them is at least the maximum of their diameters. We study the effect of the gap function for the irreducibility property on the computability of the entropy for decidable subshifts - meaning subshifts whose language is decidable. Subshifts of finite type are not necessarily in this class. However, $o(n)$-irreducible ones are. For the class of decidable subshifts, we characterize the threshold with a summability condition:

Theorem 1.3 ([GH16]). *Let $d \geq 1$ be an integer, and $f : \mathbb{N} \to \mathbb{N}$ be a computable function such that $f(0) = 0$.*

1. *If $\sum_n \frac{f(n)}{n^2}$ converges at a computable rate, there exists an algorithm that (uniformly) computes the entropy of f-irreducible decidable $\mathbb{Z}^d$-subshifts.*
2. *If $\sum_n \frac{f(n)}{n^2} = +\infty$, the set of numbers that are entropy of a decidable f-irreducible $\mathbb{Z}^d$-subshift are exactly the Π_1-computable numbers.*

The main tools used in the proof of this result are the bounded density subshifts, introduced by B. Stanley [Sta13], and defined on alphabet $\{0,1\}$ by rules which consist in forbidding for all $n \geq 1$ the number of 1 symbols in a length n word to be greater than p_n, where $(p_n)_n$ is a sequence of positive integers. We program bounded density subshifts in order to prove the second point of the theorem, and the condition on the function f serves to ensure that the entropy of the constructed subshift is the intended one. A notable aspect of the construction is that we use sequences $(p_n)_n$ coming from the discretisation of concave piecewise linear maps: this allows programming these sequences by programming only the sparse sub-sequence $(p_{F^n(1)})_n$, where $F(k) = 2k + f(k)$ for all $k \geq 1$. The block gluing condition is translated into a lower bound on the possible values that $p_{F^{n+1}(1)}$ can take, according to $p_{F^n(1)}$.

Let us note that a small gap remains in the conditions of the theorem, since in the first case the series is assumed to have computable convergence rate. Although it is not proved in this text, it seems that this condition is artificial can be removed.

The main obstacle to having a similar result for f-block gluing SFT seems to be the way hierarchical structures -used to embed computations - are constructed, preventing sub-linear block gluing.

1.3.2 Minimality

In Chapter 6, we study the properties of entropy dimension of minimal $\mathbb{Z}^d$-SFT. Recall that a $\mathbb{Z}^d$-subshift is minimal when all the configurations of this subshift share the same patterns.

It is a known fact that the entropy of a minimal SFT is zero. However, the study of another topological invariant, the entropy dimension, reveals that this class of systems is also rich, in the sense that this is possible to embed universal computation in it. The entropy dimension of a subshift X is defined as

$$D_h(X) = \lim_n \frac{\log(\log(N_n(X)))}{\log(n)},$$

when this limit exists, where $(N_n(X))_n$ is the complexity sequence of the subshift. This entropy dimension was introduced in [dC97] in order to measure the complexity of zero entropy systems, and its properties are studied in [DHKP11]. In [Mey11], T. Meyerovitch characterized the numbers that are entropy dimension of a multidimensional SFT.

The construction of T. Meyerovitch uses similar ideas as in [HM10], except that the bits are chosen using a selection process through the hierarchical structures of the subshift of [Rob71], thought as tree structures, which is similar to the construction process of a Cantor set. Random bits then generate the entropy dimension.

We used the ideas we developed in our work on the block gluing property (the simulation of degenerated behavior of the machines and the error signals) in order to adapt T. Meyerovitch's characterization to minimality. We proved the following characterization:

Theorem 1.4 ([GS17b]). *The numbers that are entropy dimension of a minimal $\mathbb{Z}^3$-SFT are the Δ_2-computable numbers in* $[0,2]$.

The difficulty of the proof comes from the fact that the simulating machines start on an initial tape which is written with a word which has no constraint: one configuration can have all its simulating machines starting on tapes written with some words, and another configuration with other words.

The idea is to use counters that alternate all the possible initial tapes in any configuration. We use the same idea in order to alternate all the possible local layouts of random bits, although for this we need a specific mechanism to increment the counters. We code these counters so that their periods at different scales are different Fermat numbers $2^{2^n}+1$, which are coprime (this is Golbach's theorem). We need this in order to ensure the minimality property.

A more detailed sketch of this proof can be found in Chapter 6, Section 6.3.2.

In the constructions that we present in Chapter 4 and Chapter 6, dynamical constraints were translated into a heuristic principle of **separation of information**. This means that information has to be localized in different parts of the structures in order to ensure the dynamical constraints, so that only coherent information can appear in the degenerated versions of these structures. For instance, in the construction of Chapter 4 the information that tells if the machine is simulating or computing can not be transported by the machine head. If this was the case, one would not be able to simulate a degenerated behavior of a machine head which transports the information that the machine is not simulating. This forces the use of signals, and in particular **error signals**. This mechanism allows a machine which ended its computation in an error state to communicate this information to its initial location where its initialization is verified. This allows the inclusion of degenerated behaviors of the machines in degenerated structures among the possible behaviors of the machines in typical structures.

1.3.3 Computable dynamical systems on computable compact metric spaces

The aim of Chapter 7 is to provide a formalization of the problem outside the world of SFT, into the wild world of computable metric spaces. In particular, we present some notion of computability which is adapted to the study of computability of growth-type invariants for dynamical systems.

We prove a general obstruction for the computability of the entropy and entropy dimensions of computable systems on computable compact metric spaces.

After this, we characterize the possible entropies of computable systems on the Cantor set, and analyze the effect of surjectivity constraint, providing the realization of all the Σ_1-computable numbers, but no characterization.

On the real unit interval, the possible entropies of computable dynamical systems are the Σ_1-computable non negative real numbers. We also prove that the entropies of computable systems on the interval under some constraint on the total variation are exactly the computable numbers. However, we don't understand the minimal sufficient conditions required to impose the computability of the entropy for these systems.

In Chapter 8, we provide comments on the previous chapters, and perspectives for further research.

Since in this text we deal with complex and sometimes counter-intuitive systems, the constructions are made as precise as possible. This results in long lists of local rules allowing the enforcement of aimed behaviors, which are hard to read. That is why there are numerous pictures illustrating these rules. We advise the reader to read the Global behavior part and the illustrating pictures before any attempt to enter in the details of the local rules. Moreover, the constructions that we describe in this text overlap on some parts. However, since the corresponding features of the constructions can not be described independently - since they are coded in a different way in different constructions - we have to present them anew in each of the constructions.

Chapter 2

Subshifts dynamical systems

Sommaire

In this chapter, we present more thoroughly the model of $\mathbb{Z}^d$-subshifts and related notions: dynamical properties, growth-type invariants, etc. This chapter is meant as an introduction to the manipulation of subshifts in relation with dynamics. Most of the results presented in this section are already known. We present them in this text so that it is as self-contained as possible.

2.1 Subshifts

2.1.1 Subspaces of the full shift

Let $\mathcal{A}$ be a finite set, that we consider as a topological space with the discrete topology. For any $d \geq 1$, the set $\mathcal{A}^{\mathbb{Z}^d}$, called the **full shift**, is a topological space with the (infinite) power of the discrete topology. The notion of convergence associated with this topology is viewed as follows: a sequence of elements of $\mathcal{A}^{\mathbb{Z}^d}$ converges towards some other point in this space when for any $n \geq 0$, there exists some k_n such that for all $k \geq k_n$, the elements of this sequence coincide on $[\![-n, n]\!]^d$. This space is **compact**.

The shift action: The group $\mathbb{Z}^d$ acts on $\mathcal{A}^{\mathbb{Z}^d}$ through the shift action σ, defined as follows for all $\mathbf{u}, \mathbf{v} \in \mathbb{Z}^d$ and $x \in \mathcal{A}^{\mathbb{Z}^d}$:

$$(\sigma^{\mathbf{u}}.x)_{\mathbf{v}} = x_{\mathbf{u}+\mathbf{v}}.$$

Definition 2.1. *A $\mathbb{Z}^d$-**subshift** is a closed subset $X \subset \mathcal{A}^{\mathbb{Z}^d}$ which is invariant under the action of the shift: for all $\boldsymbol{u} \in \mathbb{Z}^d$, $\sigma^{\boldsymbol{u}}.X \subset X$. The integer d is the dimension of the subshift.*

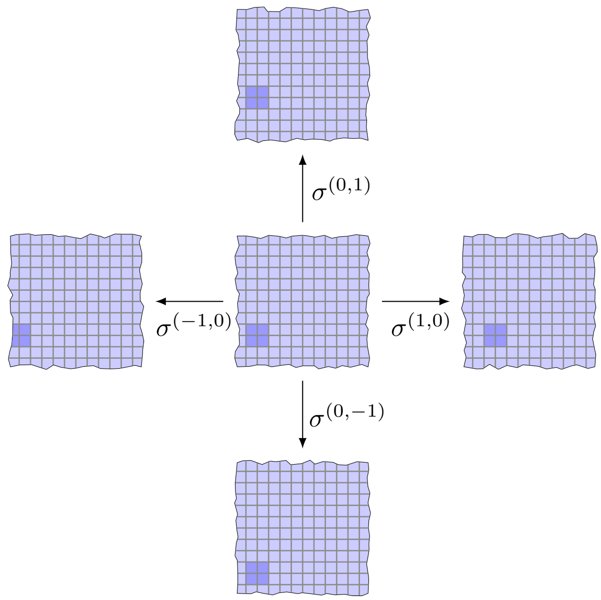


Figure 2.1: Illustration of the shift action.

A subshift is thus a topological space, with the restriction of the topology of $\mathcal{A}^{\mathbb{Z}^d}$.

Let us note that for X and Z be two $\mathbb{Z}^d$-subshifts on respective alphabets $\mathcal{A}$ and $\mathcal{D}$, the set $X \times Z$ is a $\mathbb{Z}^d$-subshift on alphabet $\mathcal{A} \times \mathcal{D}$.

2.1.2 Local aspect of subshifts

2.1.2.1 Patterns

Define $\sim$ the equivalence relation on the set of finite subsets of $\mathbb{Z}^d$ by $\mathbb{U} \sim \mathbb{U}'$ if and only if there exists $\mathbf{k} \in \mathbb{Z}^d$ such that $\mathbf{k} \in \mathbb{Z}^d$ and $\mathbb{U}' = \mathbf{k} + \mathbb{U}$. A **support** is some equivalence class for this relation.

For the purpose of notation, let us also denote $\sim$ the equivalence relation on the set of elements of some $\mathcal{A}^{\mathbb{U}}$ for $\mathbb{U}$ a finite set of $\mathbb{Z}^d$ defined by $p \sim p'$ when:

1. $p \in \mathcal{A}^{\mathbb{U}}$, $p' \in \mathcal{A}^{\mathbb{U}'}$,
2. and there exists some $\mathbf{k} \in \mathbb{Z}^d$ such that $\mathbb{U}' = \mathbf{k} + \mathbb{U}$ and $p'_{\mathbf{u}+\mathbf{k}} = p_{\mathbf{u}}$ for all $u \in \mathbb{U}$.

A **pattern** having support the equivalence class of $\mathbb{U}$ is an equivalence class for this relation.

Remark 1. *In order to simplify the exposition, we often identify a pattern with any of its elements. We also identify a support with any of its elements, as this is usually done in the literature on the subject.*

We denote $\mathbb{U}_n^{(d)}$ the set $[\![1, n]\!]^d$. A pattern on this support is called a n-**block**. We refer to a pattern on $[\![1, n_1]\!] \times ... \times [\![1, n_d]\!]$ as a $n_1 \times ... \times n_d$ pattern.

2.1.2.2 Appearance

A pattern p **appears** in an element x of $\mathcal{A}^{\mathbb{Z}^d}$ when there exists some $\mathbb{U}$ such that $x_{\mathbb{U}} = p$.

Figure 2.2: A representation of a support and a pattern p on this support.

Figure 2.3: A representation the support $\mathbb{U}_3^{(2)}$ and a 3-block.

Moreover, a n-block p is said to appear in position $\mathbf{k} \in \mathbb{Z}^d$ in the configuration x if $x_{\mathbf{k}+\mathbb{U}_n^{(d)}} = p$.

We say that a pattern p is a **sub-pattern** of some other pattern q when there is some $\mathbb{U}$ element of the support of p and $\mathbb{U}'$ in the support of q such that $\mathbb{U} \subset \mathbb{U}'$ such that the restriction of the element of q on $\mathbb{U}$ is equal to the restriction of the element of p on $\mathbb{U}$.

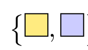
For instance, the pattern on Figure 2.2 is a sub-pattern of the pattern on Figure 2.3.

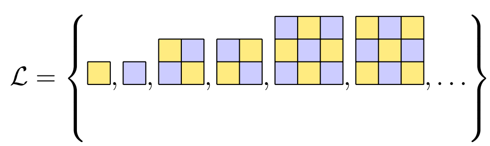
Let l, r be two integers. When a pattern p having support $[\![r, l+r]\!]^d$ is the restriction on $[\![r, l+r]\!]^d$ of another pattern q on $[\![1, l+2r]\!]^d$, we usually say that the pattern p **appears at the center** of the pattern q, or that q is centered on p.

For a $\mathbb{Z}^d$-subshift X the sets $\{x \in X : x_{\mathbb{U}} = p\}$, where p is some pattern on some support $\mathbb{U}$ that appears in an element of X, are called the **cylinders** of X. They form a basis of the topology of X. This means that every open subset of X is a union of cylinders.

2.1.3 Languages

A language on $\mathbb{Z}^d$ and on alphabet $\mathcal{A}$ is a set of patterns on alphabet $\mathcal{A}$.

Example 2.2. *For instance, the following set $\mathcal{L}$ is a language on $\mathbb{Z}^2$ and on alphabet* *:*



2.1.3.1 Subshift and languages

Definition 2.3. *The language of a subshift X is the set of patterns that appear in some element of X. We denote this set $\mathcal{L}(X)$. Its elements are called **globally admissible** patterns. For any finite subset $\mathbb{U}$ of $\mathbb{Z}^d$, we denote $\mathcal{L}_{\mathbb{U}}(X)$ the set of globally admissible patterns of X having support $\mathbb{U}$. In order to simplify the notations, when $\mathbb{U} = \mathbb{U}_n^{(d)}$, we denote this set $\mathcal{L}_n(X)$ instead, since the dimension is often non ambiguous.*

Remark 2. *The term global in the definition means that it is concerned with the whole set $\mathbb{Z}^d$. This definition contrasts with the definition of locally admissible pattern in Definition 2.6. Here the term local means that the definition is concerned by finite areas in $\mathbb{Z}^d$.*

The interest of this definition lies in the two following propositions:

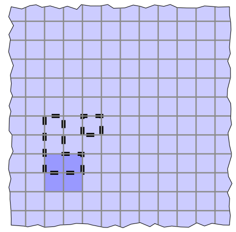


Figure 2.4: The pattern p represented on Figure 2.2 appears in some configuration x.

Proposition 2.4. *Let X be a $\mathbb{Z}^d$-subshift on some alphabet $\mathcal{A}$. Then X is the set of elements x of the full shift $\mathcal{A}^{\mathbb{Z}^d}$ such that for all n, $x_{[\![-n,n]\!]^d} \in \mathcal{L}(X)$.*

Proof. Consider a subshift X on alphabet $\mathcal{A}$.

- if x is in X, for all n, $x_{[\![-n,n]\!]^d}$ appears in x, and as a consequence is a globally admissible pattern.
- Reciprocally, if x is such that for any n, $x_{[\![-n,n]\!]^d}$ is a globally admissible pattern, then there exists a sequence $(x^{(n)})_{n\in\mathbb{N}} \in X^{\mathbb{N}}$ such that for all n,

$$x^{(n)}_{[\![-n,n]\!]^d} = x_{[\![-n,n]\!]^d}.$$

  As a consequence, the sequence $(x^{(n)})_n$ converges towards x, and since X is a closed set, x is an element of X.

□

Proposition 2.5. *Let X and Z two $\mathbb{Z}^d$-subshifts on some alphabet $\mathcal{A}$. They have the same language if and only if $X = Z$.*

Proof. If X and Z have the same language, as a consequence of Proposition 2.4, if $x \in X$, then for all n, the restriction of x to $[\![-n,n]\!]^d$ is an element of $\mathcal{L}(X) = \mathcal{L}(Z)$, so $x \in Z$. As well if $x \in Z$, then $x \in X$. This means that $X = Z$.

Reciprocally, if $X = Z$, then they have the same language. □

Definition 2.6. *Let $\mathcal{F}$ a language on $\mathbb{Z}^d$ and on alphabet A. The subshift **generated** by $\mathcal{F}$ is the set*

$$X_{\mathcal{F}} = \left\{x \in A^{\mathbb{Z}^d} : \textit{no pattern in } \mathcal{F} \textit{ appears in } x\right\}$$

*We say that the elements of $\mathcal{F}$ are forbidden. The finite patterns that have no elements of $\mathcal{F}$ as sub-pattern are called **locally admissible** patterns.*

Remark 3. *The definition of a locally admissible pattern depends on the set of forbidden patterns $\mathcal{F}$.*

Definition 2.7. *A **subshift of finite type** (SFT) is a subshift which can be generated by a finite language.*

The following proposition gives a simple normal form for subshifts of finite type:

Proposition 2.8. *Let X be some $\mathbb{Z}^d$-SFT on alphabet $\mathcal{A}$. There exists some $r > 0$ and a subset $\mathcal{F}_r$ of $\mathcal{A}^{\mathbb{U}^{(d)}_r}$ such that $X = X_{\mathcal{F}_r}$. The minimal r which verifies this is called the **rank** of the SFT X.*

Proof. Consider some finite set $\mathcal{F}$ such that $X = X_{\mathcal{F}}$, and some $r > 0$ such that all the patterns in $\mathcal{F}$ appear as sub-pattern of a pattern in $\mathcal{A}^{\mathbb{U}_r}$. Denote $\mathcal{F}_r$ the set of patterns in $\mathcal{A}^{\mathbb{U}_r}$ that have an element of $\mathcal{F}$ as a sub-pattern. We claim that $X = X_{\mathcal{F}_r}$. Indeed:

- if $x \in X$, then no sub-pattern of x can be in $\mathcal{F}$, so it has no sub-patterns in $\mathcal{F}_r$: if it had, then it would had a sub-pattern in $\mathcal{F}$.
- reciprocally, consider $x \in X_{\mathcal{F}_r}$. Assume that some element p of $\mathcal{F}$ appears in this configuration: this means that there exists some $\mathbf{k} \in \mathbb{Z}^d$ such that the restriction of x to $\mathbf{k} + \mathbb{U}$ is p, where $\mathbb{U}$ is the support of p. Then the restriction of x to $\mathbf{k} + \mathbb{U}_r^{(d)}$ is an element of $\mathcal{F}_r$, which is a contradiction.

$\square$

Informally, we call a constraint on the apparition of patterns in configuration of X - again, local means that the rule is concerned with a finite area in $\mathbb{Z}^d$ - a **local rule**. Sometimes a rule is expressed as a restriction on the patterns that can appear on some support to a subset of the possible patterns. This way, this set of patterns is enforced and the complementary set is forbidden, like in Example 2.10. The following proposition tells that any subshift is obtained by imposing a set of rules on the patterns that can appear.

Proposition 2.9. *Let X be a subshift of $\mathcal{A}^{\mathbb{Z}^d}$. There exists some language $\mathcal{F}$ such that $X = X_{\mathcal{F}}$.*

Proof. Denote $\mathcal{F} = \mathcal{L}(X)^c$. Then $X = X_{\mathcal{F}}$. Indeed, if x is in X then any pattern that appears in x is in the language of X and not in $\mathcal{F}$. As a consequence $x \in X_{\mathcal{F}}$. Reciprocally, if $x \in X_{\mathcal{F}}$, then for all n, $x_{[\![-n,n]\!]^d}$ is not in $\mathcal{F}$, and thus is in $\mathcal{L}(X)$. As a consequence of Proposition 2.4, $x \in X$. $\square$

The construction of a subshift often involves this proposition. When thinking of a subshift that exhibits some behavior, we have an informal description of the subshift in mind. To define the subshift formally, we search for a set of rules that induces this exact behavior. In order to verify that this set of rules works, we have to understand the subshift that is given by it. Here is an example for this description. This example shows in particular that local rules are similar to logical rules.

Example 2.10. *Let X be the subshift on alphabet $\{ , \}$, whose elements are like on Figure 2.5. This means that they verify that for any position $\boldsymbol{k} \in \mathbb{Z}^2$, the colors on positions $\boldsymbol{k} + (1,1)$ and $\boldsymbol{k}$ are the same. Moreover, the colors on positions $\boldsymbol{k} + (0,1)$ and $\boldsymbol{k}$ are different. This means that X is a subset of $X_{\mathcal{F}_1}$ with $\mathcal{F}_1 = \{ , , , \}$. Reciprocally, if a configuration is in the subshift $X_{\mathcal{F}_1}$, then consider any position $\boldsymbol{k}$ and assume (without loss of generality) that on this position the color is . Then the color in position $\boldsymbol{k} + (0,1)$ is , because the pattern is forbidden, and the color on the position $\boldsymbol{k} + (0,2)$ is again because the pattern is forbidden, etc. So the column of a configuration in $X_{\mathcal{F}_1}$ looks like a column on Figure 2.5. Moreover, the color on position $\boldsymbol{k}+(1,1)$ has to be because the pattern is forbidden, so the column containing this position is obtained from the first one by changing the color of every position in the column. Thus, using this argument multiple times, we see that a configuration in $X_{\mathcal{F}_1}$ is like on Figure 2.5, and $X = X_{\mathcal{F}_1}$.*

The subshift X is also obtained by forbidding the set $\mathcal{F}_2$ of 2-block that are different from or . Thus, this example shows that there can exist two different languages defining the same subshift. Moreover, the language of X is the language $\mathcal{L}$ in Example 2.2.

The following technical lemma relates globally admissible patterns to locally admissible ones:

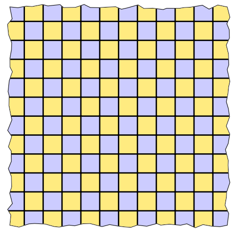


Figure 2.5: Representation of the configurations in X.

Lemma 2.11. *Let $\mathcal{F}$ be some language and denote X the subshift $X_{\mathcal{F}}$. Let u be some pattern on $[\![-n,n]\!]^d$. This pattern is globally admissible if and only if for all $N \geq n$ there exists some pattern p^N on $[\![-N,N]\!]^d$ which is locally admissible for X such that $p^N_{[\![-n,n]\!]^d} = u$.*

Remark 4. *In other words, a pattern u on $[\![-n,n]\!]^d$ is not globally admissible if and only if there exists some $N_0 \geq n$ such that no locally admissible pattern on $[\![-N_0,N_0]\!]^d$ has restriction u on $[\![-n,n]\!]^d$.*

Proof. Let u be some pattern on $[\![-n,n]\!]^d$.

- If u appears in some configuration x of X, $p^N = x_{[\![-N,N]\!]^d}$ verifies the second assertion.
- Reciprocally, assume that u verifies this second assertion. For all N there exists a configuration X^N of the full shift whose restriction on $[\![-N,N]\!]^d$ is locally admissible and the restriction on $[\![-n,n]\!]^d$ is u. These configurations form a sequence which has a convergent subsequence (since the full shift is compact). Since for all N the elements of these sequence agree on $[\![-N,N]\!]^d$ from some point, the restriction of this limit configuration is locally admissible. This means that this configuration can not contain any forbidden pattern. As a consequence, u appears in a configuration of X, and is thus globally admissible.

□

2.1.3.2 Obstructions

The following proposition exhibits restrictions on the possible languages of a subshifts.

Proposition 2.12. *A language $\mathcal{L}$ is the language of a subshift if and only if it verifies the following properties:*

1. ***Factoriality**: every sub-pattern of a pattern in $\mathcal{L}$ is in $\mathcal{L}$.*
2. ***Strict extendability:** Let p be a pattern in $\mathcal{L}$ having support $\mathbb{U}$. There exists some other block p' in $\mathcal{L}$ on support $\mathbb{U}'$, where*

$$\mathbb{U}' = \bigcup_{\epsilon_1,\epsilon_2 \in \{-1,0,1\}} ((\epsilon_1,\epsilon_2) + \mathbb{U}),$$

*and such p is a sub-pattern of p' on $\mathbb{U}$. We say that the second pattern **extends** the first one.*

Proof.
- Let $\mathcal{L}$ be the language of a subshift X, and p some pattern on support $\mathbb{U}$ that appears in the subshift. There exists some element x of X and some position $\mathbf{k} \in \mathbb{Z}^d$ such that $x_{\mathbf{k}+\mathbb{U}} = p$.

  The restriction of x on $\mathbb{U}'$ is in the language $\mathcal{L}$ and this pattern extends p. We proved that the language $\mathcal{L}$ is extendable. It is factorial because any sub-pattern of p appears also in the configuration x and so is in the language of X.

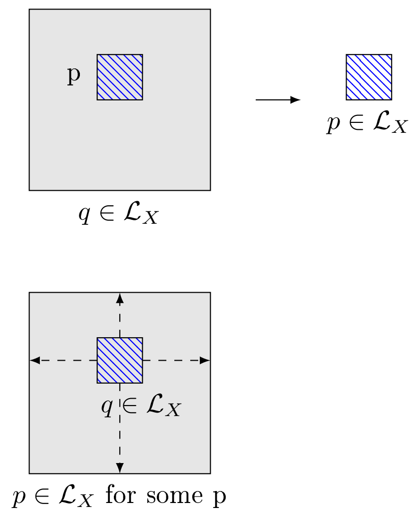
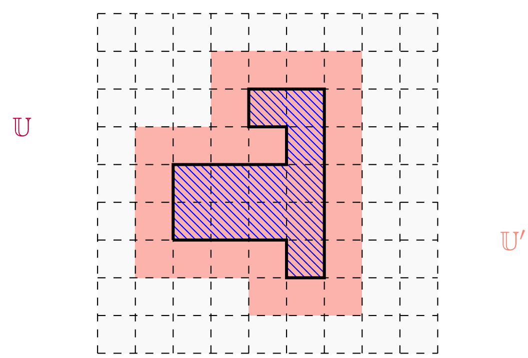


Figure 2.6: Illustration of the factoriality property (above) and the strict extendability property (under) of a language



- Let $\mathcal{L}$ be an extendable and factorial language and denote $X = X_{\mathcal{L}^c}$. Let us prove that the language of X is $\mathcal{L}$. Any pattern that appears in X has to be in $\mathcal{L}$ by definition.

  Now consider some pattern p in $\mathcal{L}$, on support $\mathbb{U}$.

  1. **The pattern p can be completed into a block:** Using the notation of the first point, let us consider the sequence of subsets of $\mathbb{Z}^d$ $(\mathbb{U}^{(m)})_m$ such that $\mathbb{U}^{(0)} = \mathbb{U}$ and for all m,

     $$\left(\mathbb{U}^{(m)}\right)' = \mathbb{U}^{(m+1)}.$$

     From the extendability of the language $\mathcal{L}$, there exists a sequence of patterns $(p^m)_m$ such that:

     (a) $p^0 = p$,

     (b) for all m, p^{m+1} extends p^m on $\mathbb{U}^{(m+1)}$,

     (c) and this pattern is in $\mathcal{L}$.

     Let us consider the configuration x whose restriction on $\mathbb{U}^{(m)}$ is p^m for all m.

2. **The configuration x is in X:** The patterns that appear in this configuration are all a sub-pattern of p^m for some m. Since $\mathcal{L}$ is factorial, this implies that this pattern is also in $\mathcal{L}$. As a consequence, x is in $X_{\mathcal{L}^c}$, and p is in the language of this subshift.

□

2.1.3.3 One dimensional SFT

The first notable effect of geometry is the gap between one-dimensional and multidimensional SFT. The particularity of one-dimensional SFT is that they have regular language. This means that the language is accepted by a finite deterministic automaton. See [HU06] for a reference on automata. More details on this part can be found in [LM95].

Let us first provide some definitions.

Definition 2.13. *A **(deterministic) finite automaton** is the gathering of some finite sets $\mathcal{Q}$ and $\mathcal{A}$, a function $\delta : \mathcal{Q} \times \mathcal{A} \to \mathcal{Q}$, and subset $\mathcal{F} \subset \mathcal{Q}$.*

A finite automaton can be represented by a finite oriented graph, whose vertices are the elements of $\mathcal{Q}$, and there is an arrow from state q to state q' if and only if there exists some $a \in \Sigma$ such that $\delta(q, a) = q'$.

Definition 2.14. *We say that a language $\mathcal{L}$ on $\mathbb{Z}$ and on alphabet Σ is **accepted** by an automaton $(\mathcal{Q}, \Sigma, \delta, \mathcal{F}, q_0)$ when a word $w = w_1...w_k$ is in $\mathcal{L}$, if and only if there exists some sequence $(q_l)_{l=1..k} \in \mathcal{Q}^k$ such that for all $l \leq k-1$, $w_{l+1} = \delta(q_l, w_l)$, and for all $l \leq k$, $q_l \notin \mathcal{F}$.*

Remark 5. *This definition means that the words of $\mathcal{L}$ are exactly the words of possible colors of a finite path on the graph of the automaton.*

Theorem 2.15 ([LM95])**.** *Let X be a $\mathbb{Z}$-SFT on some alphabet $\mathcal{A}$. There exists a finite automaton such that the language of X is accepted by some deterministic finite automaton.*

Remark 6. *This means that the elements of X are the words of colors of biinfinite paths on the graph of the automaton.*

Proof. Let $r > 0$ be the rank of the SFT. Then $X = X_{\mathcal{F}_0}$ for $\mathcal{F}_0$ some subset of the r blocks on the alphabet $\mathcal{A}$. Let $\Sigma = \mathcal{A}$ and $\mathcal{Q} = A^{[\![1,r]\!]}$, $F = \mathcal{F}_0$, and for all $q = q_1...q_r$ and $a \in \mathcal{A}$, $\delta(q, a)$ is $q_2...q_r a$. □

Remark 7. *It would be interesting to understand more precisely how the unique dimension influences information processing in subshifts of finite type, and moreover if there is a difference from this point of view between higher dimensions.*

2.2 Dynamics

In this section we present dynamical aspects of subshifts.

2.2.1 Appearance properties

In the context of subshifts, widely studied dynamical properties like minimality or transitivity have a translation in terms of appearance in configurations of the subshift.

Definition 2.16. *A $\mathbb{Z}^d$-subshift X is **minimal** when it does not have a proper subshift. This means that Z is a non-empty subshift included in X, then $Z = X$.*

Proposition 2.17. *Let X be a $\mathbb{Z}^d$-subshift. The following conditions are equivalent:*

1. *X is minimal.*
2. *For all x element of X, and a non-empty open set U of the subshift, there exists some $\mathbf{u} \in \mathbb{Z}^d$ such that $\sigma^{\mathbf{u}}.x \in U$.*
3. *Any pattern in the language of X appears in all the configurations of X.*

Proof.

- $1 \Leftrightarrow 2$. Assume that X is minimal. For all x, the set

$$\overline{\{\sigma^{\mathbf{u}}(x) : \mathbf{u} \in \mathbb{Z}^d\}}$$

is a closed set and invariant under the shift action. Since X is minimal, this set is equal to X. This exactly means that the sequence $(\sigma^{\mathbf{u}}(x))_{\mathbf{u}\in\mathbb{Z}^d}$ is dense in X. As a consequence, for all non-empty open set U of the subshift, there exists some $\mathbf{u} \in \mathbb{Z}^d$ such that $\sigma^{\mathbf{u}}.x \in U$.

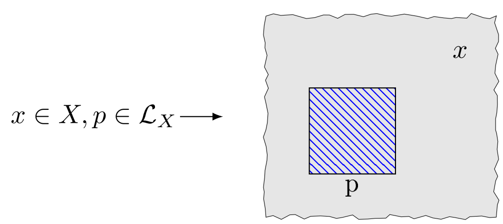
Reciprocally, let Z be some non-empty subshift such that $Z \subset X$. This means that there exists some $x \in Z$. Since Z is a subshift,

$$\overline{\{\sigma^{\mathbf{u}}.x : \mathbf{u} \in \mathbb{Z}^d\}} \subset Z.$$

Since X is minimal and in particular $x \in X$, the set on the left is equal to X. Thus $X \subset Z \subset X$. This means that $Z = X$.

- $2 \Leftrightarrow 3$. Since the cylinders are a basis of the topology of X, the condition 2 is equivalent to the condition that for all x and for all cylinder U of the subshift, there exists some $\mathbf{u} \in \mathbb{Z}^d$ such that $\sigma^{\mathbf{u}}.x \in U$. Since a cylinder is a set $\{x : x_{\mathbb{U}} = p\}$, for $\mathbb{U}$ some finite subset of $\mathbb{Z}^d$, and p a pattern in the language of X, the relation $\sigma^{\mathbf{u}}.x \in U$ means that p appears in x on position $\mathbf{u}$. As a consequence, condition 2 is equivalent to the condition that any pattern in the language of X appears in all the configurations of this subshift - this is condition 3.

□



Definition 2.18. *A $\mathbb{Z}^d$-subshift X is* ***transitive*** *when for any ordered pair (U, V) of two non-empty open sets of X, there exists some $\mathbf{u} \in \mathbb{Z}^d$ such that $\sigma^{\mathbf{u}}.U \cap V \neq \emptyset$.*

Proposition 2.19. *A $\mathbb{Z}^d$-subshift is transitive if and only if for any ordered pair of patterns (p, q) in the language of X, the two patterns appear in some configuration $x_{p,q}$ of X.*

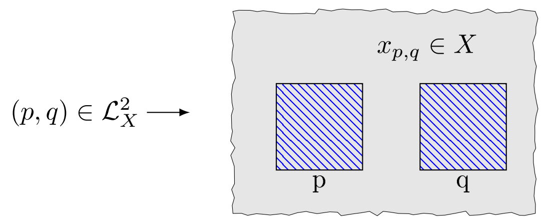
Proof. For $\mathbf{u} \in \mathbb{Z}^d$,

$$U = \{x \in X : \ x_{\mathbb{U}} = p\}$$

and

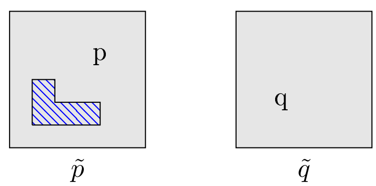
$$V = \{x \in X : \ x_{\mathbb{V}} = q\}$$

two cylinders of X, the formula $\sigma^{\mathbf{u}}.U \cap V \neq \emptyset$ means that there exists some configuration x such that p appears on $\mathbb{U}$ in x and q appears on $\mathbf{u} + \mathbb{V}$ in x. □



Remark 8. *If a subshift is minimal, then it is transitive.*

Proposition 2.20. *A $\mathbb{Z}^d$ subshift X is transitive (resp. minimal) if and only if for any ordered pair of blocks in its language having the same size, the two blocks appear in a same (resp. any) configuration x in X.*

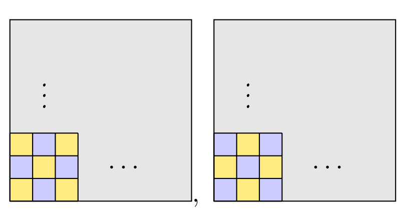


Proof. We prove this proposition only for the transitivity property, since the proofs of the two statements are similar.

- If a subshift verifies this property, then it is transitive by Proposition 2.19.
- Reciprocally, let X be a subshift such that for any ordered pair of blocks in the language of X, the two blocks appear in some configuration x of X. Consider two patterns p, q in the language of X. For any finite set of $\mathbb{Z}^d$ there exists some integer n_0 such that for all $n \geq n_0$, this set is included in $[\![-n, n]\!]^d$. As a consequence, p and q can be extended into blocks having the same size, denoted respectively $\tilde{p}$ and $\tilde{q}$, Since $\tilde{p}$ and $\tilde{q}$ appear in some configuration x, this implies that p and q appear in this same configuration. Hence the subshift is transitive.

□

Example 2.21. *The subshift presented in Example 2.10 is minimal. Indeed for any n there are exactly two n-blocks in the language of this subshift. One has a gold color on the bottom leftmost position and one with blue on this position:*

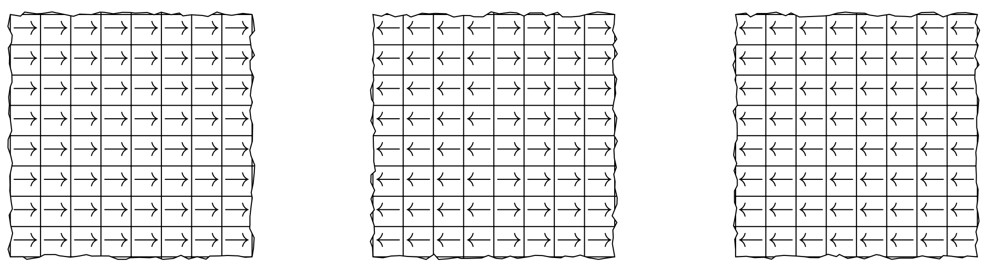


These two blocks appear in any configuration of the subshift, from its description. This means, by Proposition 2.20, that the subshift is minimal, and by Remark 8, it is also transitive.

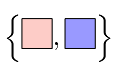
Example 2.22. *The subshift on alphabet* $\left\{ \boxed{\rightarrow}, \boxed{\leftarrow} \right\}$ *given by forbidding the patterns*

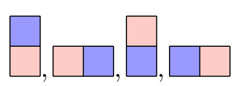
$$\boxed{\rightarrow}\boxed{\leftarrow}\,, \quad \begin{array}{c}\boxed{\leftarrow}\\ \boxed{\rightarrow}\end{array}, \quad \begin{array}{c}\boxed{\rightarrow}\\ \boxed{\leftarrow}\end{array}$$

is transitive, but not minimal. Consider some configuration of this subshift. It contains either the symbol $\boxed{\rightarrow}$ *or* $\boxed{\leftarrow}$. *Assume that it contains* $\boxed{\rightarrow}$. *The two last rules of the set of rules imply that on the top and bottom positions of the position where this symbol appears, the same symbol appears. This means, by a recursion argument, that the whole column contains this symbol. Moreover, the symbol on the right position of it is also* $\boxed{\rightarrow}$, *using the first rule. With the two last rules, this means that the whole column on the right is full of* $\boxed{\rightarrow}$, *and by a recursion argument, the right half plane is full of* $\boxed{\rightarrow}$. *Now there are two cases: the configuration contains also a left arrow, or not. In the first case, the configuration consists of two vertical half planes, the one on the left being full of left arrows, and the other one of right arrows. In the second case, the plane is full of right arrows. As a consequence, considering the other possibility left that the plane is full of left arrows, all the configurations in this subshift look as follows.*



For any n, *the* n*-blocks that appears in the language of this subshift appear some configuration (in fact any) configuration corresponding to the second picture. This means that the subshift is transitive. However it is not minimal: some patterns in the second type configuration don't appear in the first type configurations (the pattern* $\boxed{\leftarrow}$ *for instance).*

Example 2.23. *The pattern on the alphabet*  *with forbidden patterns*



is not transitive. Indeed, the subshift consists of two configurations: one fully salmon and one fully blue. Hence a fully blue block can not appear in a configuration with a fully salmon block.

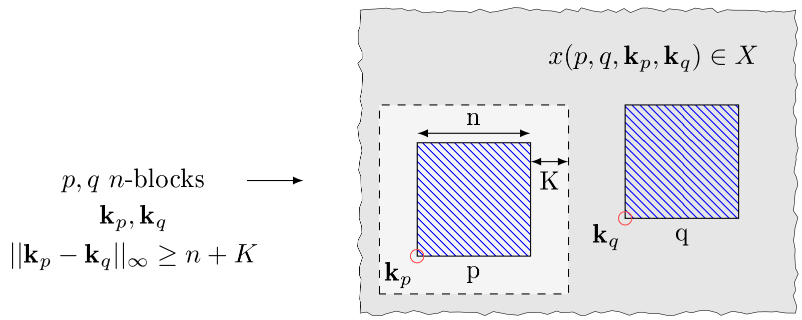
2.2.2 Variations

Recent works studied variations of these fundamental appearance properties, introducing restrictions on the distance between the patterns. With this modification, the shape of the patterns has an influence, and properties related to restricted forms of patterns to be glued appeared. We present in this section some of these properties that are gathered in [BPS10], and reproduce the proof that these properties define different classes of subshifts. In Chapter 4, we will define other variations of these properties.

Definition 2.24. *Let $K > 0$ a positive real number. A $\mathbb{Z}^d$ subshift X is K-block gluing when for any integer n and any pair of n-blocks p, q in its language, the two blocks appear in any positions $\boldsymbol{k}_p$ and $\boldsymbol{k}_q$ such that the distance between these positions is greater than $K + n$:*

$$||\boldsymbol{k}_p - \boldsymbol{k}_q||_\infty \geq K + n,$$

in some configuration $x(p, q, \boldsymbol{k}_p, \boldsymbol{k}_q)$ in X.



For U a finite subset of $\mathbb{Z}^d$, we denote $\delta(U)$ its diameter. It is defined as the maximal distance between two of its elements. We also denote, by abuse of notation, for two finite subsets $\mathbb{U}, \mathbb{V}$ of $\mathbb{Z}^d$,

$$\delta(\mathbb{U}, \mathbb{V}) = \min_{\mathbf{k} \in \mathbb{U}} \min_{\mathbf{l} \in \mathbb{V}} d(\mathbf{k}, \mathbf{l}).$$

Definition 2.25. *Let $K > 0$ a positive real number. A $\mathbb{Z}^d$-subshift X is K-strongly irreducible when for any two finite subsets $\mathbb{U}$ and $\mathbb{V}$ of $\mathbb{Z}^d$ such that*

$$\delta(\mathbb{U}, \mathbb{V}) \geq K,$$

and for all p, q patterns on respective support $\mathbb{U}, \mathbb{V}$ in $\mathcal{L}(X)$, there exists $x(p, q)$ in X whose restrictions on $\mathbb{U}$ and $\mathbb{V}$ are respectively p and q.

Remark 9. *Consider two integers k, l such that $l > 2k > 0$. We denote $\mathbb{O}_{k,l}$ the finite set of $\mathbb{Z}^d$ defined by:*

$$\mathbb{O}_{k,l} = [\![0, l-1]\!]^d \backslash [\![k-1, l-k-1]\!]^d.$$

If a $\mathbb{Z}^d$-subshift X is K-strongly irreducible, in particular we have the following. For any ordered pair of integers (n, m), any n-block p and any pattern q having support $\mathbb{O}_{2(K+m)+n,m}$, there exists some configuration $x(p, q) \in X$ such that the restriction of $x(p, q)$ on $\mathbb{O}_{2(K+m)+n,m}$ is q and its restriction on $[\![K + m, K + m + n - 1]\!]^d$ is p. See an illustration on Figure 2.7.

The following proposition states that the classes of subshifts defined by these properties are comparable, although different. Moreover, it exhibits an effect of the dimension on classes of subshifts defined by a dynamical property.

Proposition 2.26. *Let X be a $\mathbb{Z}^d$-subshift. If $d \geq 2$, then if X is K-irreducible, then it is K-block gluing. This implication can not be reversed, even when the considered subshift is a SFT.*

Remark 10. *When $d = 1$, the implication can be reversed. This is proved in Chapter 4 for irreducible subshifts with gap function.*

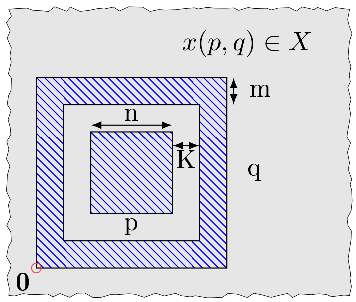


Figure 2.7: A particular case of the K-strong irreducibility property.

Proof. 1. Let us first prove the implication.

Let $d \geq 2$, and assume that X is K-strongly irreducible. Consider $n \geq 0$ and p, q two n-blocks in $\mathcal{L}(X)$, and $\mathbf{k}_p, \mathbf{k}_q \in \mathbb{Z}^2$ such that $||\mathbf{k}_p - \mathbf{k}_q|| \geq n + K$. For m large enough, we can pick some $\mathbf{u} \in \mathbb{Z}^2$ such that $\mathbf{u} \in \mathbb{O}_{2(K+m)+n,m}$, and $\mathbf{u} - (m+K, ..., m+K) = \mathbf{k}_q - \mathbf{k}_p$. The pattern q placed on position $\mathbf{u}$ can be completed into a pattern q' on $\mathbb{O}_{2(K+m)+n,m}$. According to the K-irreducibility property, there exists some configuration $x(p, q') \in X$ such that q' appears on $\mathcal{O}_{2(K+m)+n,m}$ and p on $(m+K, ..., m+K) + [\![0, n-1]\!]^d$. See Figure 2.8.

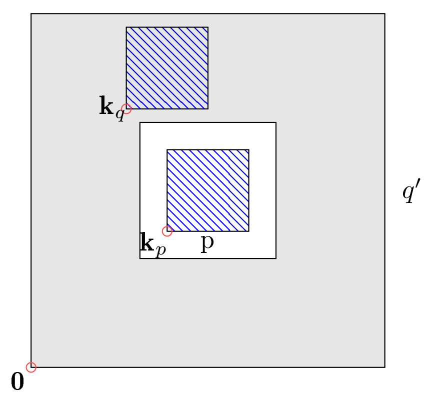
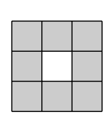


Figure 2.8: Illustration of the proof of Proposition 2.26.

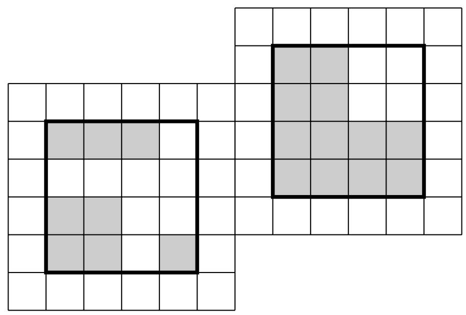
This proves the K-block gluing property.

2. Let us present a simple example of 2-block gluing $\mathbb{Z}^2$ subshift which is not strongly irreducible (this can be adapted without great difficulties into $\mathbb{Z}^d$-subshifts with $d \geq 3$). The alphabet of this subshift is $\{\blacksquare, \square\}$, and it is defined by imposing that whenever a l-block is such that for all $\mathbf{u} \in \mathbb{O}_{1,l}$, $p_{\mathbf{u}} = \blacksquare$, then for all $\mathbf{u} \in [\![0, l-1]\!]^2$, $p_{\mathbf{u}} = \blacksquare$. For instance, the following pattern is forbidden:

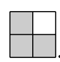


Any all-white block is in the language of X, and the pattern on any $\mathbb{O}_{1,l}$ which is colored all gray is also in the language of X. However the first one never appears 'inside' of the other one. Hence this subshift can not be K-strongly irreducible for any $K > 0$. However, it is 2-block gluing.

Indeed, consider two n-blocks for some n. These blocks can be completed with white symbols into $n+2$-blocks, because it does not make 'hole' patterns appear. For the same reason, we can place them into any two positions, without overlap, and complete it into a configuration of X by filling the plane with white symbols, as on Figure 2.



However, this subshift is not a priori a SFT. To construct a SFT which is 2-block gluing but not strongly irreducible, one can consider the subshift on the alphabet {■,□}, and defined by the forbidden pattern:

.

This subshift, presented in [BPS10], exhibits the same **behavior** as the first one - meaning that a ring colored all gray implies that the inside of it is also all gray. Consider such an annulus, illustrated on Figure 2.9. The coloring of the bottom left corner of the ring implies that on the position in the red square left undefined, the symbol is ■. Same argument applies to the picture below, and with a recursion argument, the inside of the pattern has to be colored all gray.

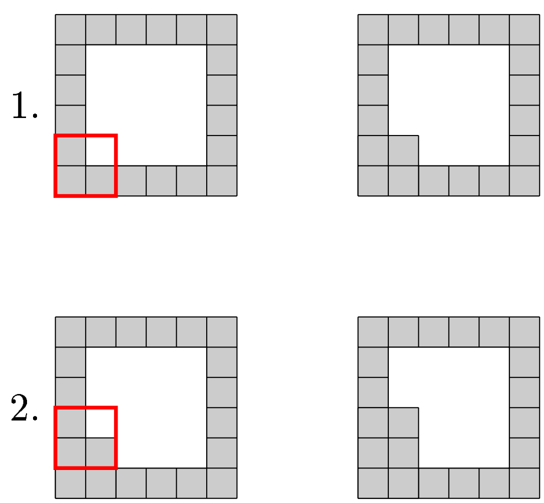


Figure 2.9: Illustration of the SFT rule

For the same reasons as above, this second subshift is also 2-block gluing and not strongly irreducible.

□

2.2.3 Morphisms of subshifts

2.2.3.1 Block maps

There is a natural definition for a morphism of subshifts:

Definition 2.27. *A morphism between two $\mathbb{Z}^d$-subshifts X, Z is a continuous map $\varphi : X \to Z$ such that $\varphi \circ \sigma^{\boldsymbol{v}} = \sigma^{\boldsymbol{v}} \circ \varphi$ for all $\boldsymbol{v} \in \mathbb{Z}^d$ (the map commutes with the shift action).*

These morphisms are characterized by a local counterpart:

Theorem 2.28 ([Hed69])**.** *Let X, Z be two $\mathbb{Z}^d$ subhifts on respective alphabets $\mathcal{A}_X$ and $\mathcal{A}_Z$. A function $\varphi : X \to Z$ is a morphism if and only if there exists some $K > 0$ and some*

$$f : \mathcal{A}_X^{[\![-K,K]\!]^d} \to \mathcal{A}_Z$$

such that for all $\boldsymbol{v} \in \mathbb{Z}^d$, and $x \in X$,

$$\varphi(x)_{\boldsymbol{v}} = f(x_{\boldsymbol{v}+[\![-K,K]\!]^d}).$$

A function verifying this condition is called a K-block map.

Proof. Consider such a K-block map φ.

- Let us first prove that the function φ is continuous. This is sufficient to prove that for any cylinder U, the $\varphi^{-1}(U)$ is an open set, since any open set is a union of cylinders. Let U be some cylinder: there exists some finite set $\mathbb{U} \in \mathbb{Z}^d$ and a pattern p on $\mathbb{U}$ such that U is the set of configurations x such that the restriction on $\mathbb{U}$ of x is p. Consider the set:

$$\mathbb{U}' = \bigcup_{\mathbf{u} \in \mathbb{U}} \left(\mathbf{u} + [\![-K, K]\!]^d\right),$$

  which is a finite subset of $\mathbb{Z}^d$. It is sufficient to know x on $\mathbb{U}'$ in order to know $\varphi(x)$ on $\mathbb{U}$, because φ is a K-block map. This means that $\varphi^{-1}(U)$ consists in a finite union of cylinders. The set $\varphi^{-1}(U)$ is described as the set of configurations whose restriction on $\mathbb{U}'$ is the set of restrictions of a configuration x to $\mathbb{U}'$ such that $\varphi(x)$ has restriction p on $\mathbb{U}$. This means that φ is continuous.

- Consider some configuration x in X and some vectors $\mathbf{u}, \mathbf{v} \in \mathbb{Z}^d$. Then

$$\varphi \circ \sigma^{\mathbf{u}}(x)_{\mathbf{v}} = f(\sigma^{\mathbf{u}}(x)_{\mathbf{v}+[\![-K,K]\!]^d})$$

  and $\sigma^{\mathbf{u}}(x)_{\mathbf{v}'} = x_{\mathbf{u}+\mathbf{v}'}$ for all $\mathbf{v}'$. Hence

$$\varphi \circ \sigma^{\mathbf{u}}(x)_{\mathbf{v}} = f(x_{\mathbf{v}+\mathbf{u}+[\![-K,K]\!]^d})$$

  For the same reason, we have that

$$\sigma^{\mathbf{u}} \circ \varphi(x)_{\mathbf{v}} = f(x_{\mathbf{v}+\mathbf{u}+[\![-K,K]\!]^d})$$

  As a consequence, $\varphi \circ \sigma^{\mathbf{u}} = \sigma^{\mathbf{u}} \circ \varphi$ for all $\mathbf{u} \in \mathbb{Z}^d$.

Consider then some morphism ψ.

For all $\mathbf{u} \in \mathbb{Z}^d$ and $a \in \mathcal{A}_Z$, we

$$U_{\mathbf{u},a} = \{x \in X,\ \psi(x)_{\mathbf{u}} = a\}.$$

Since the space $\mathcal{A}_X^{\mathbb{Z}^d}$ is compact and $U_{\mathbf{u},a}$ is a compact subset, $\psi^{-1}(U_{\mathbf{u},a})$ is also a compact subset. Since the alphabet is finite, the set

$$\bigcup_{a \in \mathcal{A}_Z} \psi^{-1}(U_{\mathbf{u},a})$$

is also a compact subset, and as a consequence is included in a finite union of cylinders. This means that there exists some $K > 0$ such that for all $a \in \mathcal{A}_Z$, $\psi^{-1}(U_{\mathbf{u},a})$ is the set of configurations whose restriction on $\mathbf{u} + [\![-K, K]\!]^d$ is in some subset of patterns of $\mathcal{A}_Z^{[\![-K,K]\!]^d}$. Let $f_{\mathbf{u}}$ the function which to a pattern p on support $\mathbf{u} + [\![-K, K]\!]^d$ associates the letter a such that the set of configurations whose restriction to $\mathbf{u} + [\![-K, K]\!]^d$ is p is included in $\psi^{-1}(U_{\mathbf{u},a})$. Since ψ commutes with the shift action, $f_{\mathbf{u}}$ does not depend on $\mathbf{u}$, and we denote f this function. As a consequence, ψ is a K-block map corresponding to the local function f.

$\square$

Definition 2.29. *A morphism $\psi : X \to Z$ is called a* ***conjugacy*** *when there exists some morphism φ such that $\varphi \circ \psi = id_X$ and $\psi \circ \varphi = id_Z$.*

Corollary 2.30. *A morphism is a conjugacy if and only if it is an invertible map.*

Proof. It is clear that a conjugacy is an invertible map. Reciprocally, considering an invertible morphism ψ, its inverse map is shift commuting and continuous, by compactness. Hence this is a morphism, and thus ψ is a conjugacy. $\square$

2.2.3.2 Projection over a layer

Subshifts are often constructed on an alphabet $\mathcal{A}$ which is the product of some smaller alphabets $\mathcal{A}_i$, $i = 1..k$:

$$\mathcal{A} = \mathcal{A}_1 \times ... \times \mathcal{A}_k.$$

Consider some subshift X on this alphabet, and π_i defined applying to each letter the projection

$$\begin{array}{ccc} \mathcal{A}_1 \times ... \times \mathcal{A}_k & \to & \mathcal{A}_i \\ (a_1, ..., a_k) & \mapsto & a_i \end{array}$$

The function π_i is a morphism from X to ${\mathcal{A}_i}^{\mathbb{Z}^d}$. According to Proposition 2.31, the image of φ_i is a subshift. This subshift is called the ith **layer** of X, and the function φ_i is called the projection morphism on the ith layer.

Proposition 2.31. *Let $\varphi : X \to Z$ be a morphism between two $\mathbb{Z}^d$-subshifts. The set $\varphi(X)$ is a $\mathbb{Z}^d$-subshift.*

Proof. Since φ is a continuous function and X is a compact space, then $\varphi(X)$ is a compact space. Moreover, since φ commutes with the shift action and X is stable under this action, then $\varphi(X)$ is also stable under the shift. $\square$

In the representations of configurations in a subshift, we often superimpose the alphabets, as on Figure 2.10. The figure represents the projections of a configuration in the full shift on alphabet $\{\square, \blacksquare\} \times \{0, 1, \square\}$. over its two layers.

The symbol $\square$ is called blank symbol. When superimposed on other symbols, it is not represented. It means the absence of information.

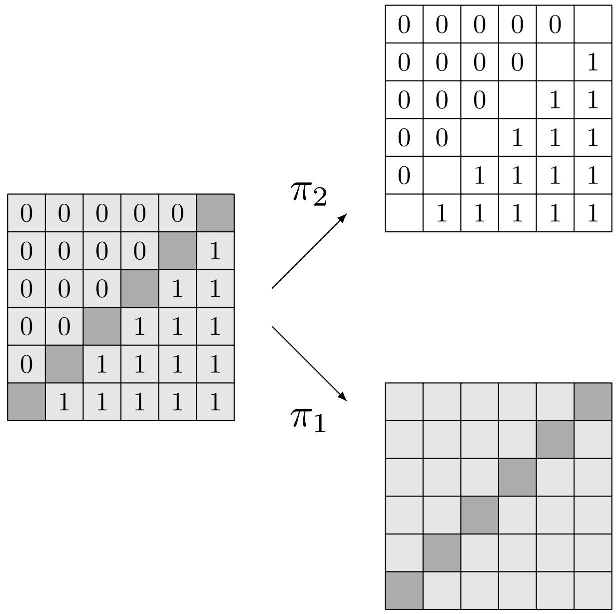


Figure 2.10: Illustration of the layer projections definition.

2.2.3.3 Effect on dynamical properties

The following proposition states that minimality property is preserved by morphism. A similar proposition will be stated in Chapter 4 for an extension of the block gluing property introduced in this chapter.

Proposition 2.32. *Let $\varphi : X \to Z$ be a morphism between two $\mathbb{Z}^d$-subshifts. If X is minimal, then Z is minimal. In particular, if X and Z are isomorphic, then X is minimal if and only if Z is minimal.*

Proof. The morphism φ is a K-block map for some K. Let u be some $(2n+1)$-block in the language of Z. There exists v some $(2n+2K+1)$-block in the language of X whose image by the block map is v. There exists a configuration x in X such that v appears in x. As a consequence, the pattern u appears in $\varphi(x) \in Z$. Hence Z is minimal. $\square$

2.2.4 Growth-type invariants

For X a $\mathbb{Z}^d$-subshift and $k \geq 0$, we denote $N_k(X)$ the number of k-blocks in the language of X.

A **growth-type invariant** is a function $\overline{I}_\gamma$ or $\underline{I}_\gamma$ on the set of subshifts on some alphabet $\mathcal{A}$, where $\gamma : (\mathbb{N}^*)^2 \to \mathbb{R}$ and

$$\overline{I}_\gamma(X) = \limsup_k \gamma(N_k(X), k),$$

or

$$\underline{I}_\gamma(X) = \liminf_k \gamma(N_k(X), k),$$

and such that for two isomorphic subshifts X and Z, $I(X) = I(Z)$ - that is why it is called an invariant. It is called growth type invariant for the reason that it depends on the asymptotic growth of the sequence $(N_k(X))_k$.

The following proposition gives a class of functions γ such that the functions above are invariants, including the ones studied in the literature.

Proposition 2.33. *Let f be a non-decreasing function, and g some function such that for all $K > 0$,*

$$\lim_n \frac{g(n+K)}{g(n)} = 1.$$

Denote γ the function defined by

$$\gamma(m,n) = \frac{f(m)}{g(n)}.$$

The two functions $\overline{I}_\gamma$ and $\underline{I}_\gamma$ are growth-rate invariants.

Proof. Let X and Z two $\mathbb{Z}^d$ subshifts and $\varphi : X \to Z$ an isomorphism. This means that there exists some $K > 0$ such that φ and φ^{-1} are K-block maps.

1. **Comparing the growth of $(N_k(X))_k$ and $(N_k(Z))_k$:**

   For any $n > 0$, considering some n-block p in the language of Z, there exists some $(n+K)$-block q in the language of X whose image by the block map of φ is p. Because the block map is injective, this means that the number of $n+K$ block in the language of X is greater than the number of n-blocks in the language of Z. Then

   $$N_n(Z) \leq N_{n+K}(X).$$

   Using the same argument on the block map of φ^{-1}, we get

   $$N_{n-K}(X) \leq N_n(Z)$$

   when $n > K$.

2. **Consequence for the growth of the sequences $(\gamma(N_k(X),k))_k$ and $(\gamma(N_k(Z),k))_k$:**

   Because f is non-decreasing, applying this function to previous inequality,

   $$f(N_{n-K}(X)) \leq f(N_n(Z)) \leq f(N_{n+K}(X)).$$

   Then
   $$\frac{g(n-K)}{g(n)} \frac{f(N_{n-K}(X))}{g(n-K)} \leq \frac{f(N_n(Z))}{g(n)} \leq \frac{g(n+K)}{g(n)} \frac{f(N_{n+K}(X))}{g(n+K)}.$$

   This means that

   $$\frac{g(n-K)}{g(n)}\gamma(N_n(Z),n) \leq \frac{g(n+K)}{g(n)}\gamma(N_{n+K}(X),n+K).$$

3. **Invariance under conjugacy:**

   Using the property of the function g, and taking the lim inf or the lim sup of these inequalities, we get that $\overline{I}_\gamma(X) = \overline{I}_\gamma(Z)$ and $\underline{I}_\gamma(X) = \underline{I}_\gamma(Z)$. This means that $\overline{I}_\gamma$ and $\underline{I}_\gamma$ are indeed growth-type invariants.

□

Remark 11. *The set of possible values of these topological invariants is related to the possible sequences $(N_n(X))_n$, and hence to the set of possible languages that can have a subshift. Proposition 2.12 states that there are restrictions on these possible languages which imply restrictions on the topological invariants.*

2.2.4.1 Topological entropy

The most widely studied of these growth-type invariants is the entropy. Consider the function f, g such that for all n, $f(n) = \log_2(n)$ and $g(m) = m^d$. These functions verifies the conditions of Proposition 2.33, and the corresponding invariants are equal [AKM65]. For all X, the number $\overline{I}_\gamma(X) = \underline{I}_\gamma(X)$ is called the **topological entropy** of X, and is denoted $h(X)$:

$$h(X) = \lim_k \frac{\log_2(N_k(X))}{k^d} = \inf_k \frac{\log_2(N_k(X))}{k^d}.$$

Heuristically, it measures the exponential growth rate of $N_k(X)$ when it grows exponentially. When the sequence $(N_k(X))$ grows sub-exponentially, its topological entropy is zero. For instance, when X is the full shift on alphabet $\mathcal{A}$, then for all k, $N_k(X) = |\mathcal{A}|^{k^d}$. Thus the entropy of X is $\log_2(|\mathcal{A}|)$.

Example 2.34. *The* ***Golden mean shift*** *is the $\mathbb{Z}$-subshift Z on the alphabet $\{0,1\}$ defined by the set of forbidden patterns $\{11\}$. It is known that its entropy is $\frac{1+\sqrt{5}}{2}$ [LM95].*

Its two-dimensional equivalent, named ***hard square shift****, is not as well understood as this one. This is the $\mathbb{Z}^2$-SFT defined on alphabet $\{0,1\}$ and defined by forbidding the patterns*

$$1 \quad 1 \ , \quad \begin{matrix} 1 \\ 1 \end{matrix} \ .$$

R. Pavlov proved [Pav12] that the entropy of this subshift can be approximated algorithmically with exponential convergence rate, using methods inspired from percolation. However, no closed form is known for this number.

In Chapter 4 we give a closed form for the entropy of subshifts in a very simple class of two-dimensional SFT.

Remark 12. *The entropy of a subshift is equal to its Hausdorff dimension [Sim15]. This, amongst others facts, enlightens the direct relation there exists between topology and dynamics while considering subshifts.*

Proposition 2.35. *Consider some $d \geq 1$. Let X and Z be two $\mathbb{Z}^d$-subshifts. The subshift $X \times Z$ has entropy*

$$h(X \times Z) = h(X) + h(Z).$$

Proof. For all $n \geq 1$, we have the following equality:

$$N_n(X \times Z) = N_n(X) N_n(Z).$$

As a consequence,

$$\frac{\log_2(N_n(X \times Z))}{n^d} = \frac{\log_2(N_n(X))}{n^d} + \frac{\log_2(N_n(Z))}{n^d}.$$

Taking the limit, we have

$$h(X \times Z) = h(X) + h(Z).$$

□

2.2.4.2 Entropy dimensions

Other known invariants are the upper (limsup, denoted $\overline{D_h}$) and lower (liminf, denoted $\underline{D_h}$) entropy dimensions, obtained by taking $f(n) = \log_2(\log_2(n))$ and $g(m) = \log_2(m)$ for all $n, m > 0$. When $\overline{I}_\gamma(X) = \underline{I}_\gamma(X)$, we denote D_h the limit, which is called the entropy dimension. The existence of this limit and its value are topological invariants. These invariants were introduced in [DHKP11] in order to measure the sub-exponential complexity of zero entropy dynamical systems. One can find there proofs of the elementary properties of these invariants.

Proposition 2.36. *The two entropy dimensions of a $\mathbb{Z}^d$-subshift are smaller than d.*

Proof. Let X be a $\mathbb{Z}^d$-SFT on some alphabet $\mathcal{A}$. For all n, $N_n(X) \leq |\mathcal{A}|^{n^d}$. This implies that

$$\frac{\log_2(\log_2(N_n(X)))}{\log_2(n)} \leq d + \frac{\log_2(\log_2(|\mathcal{A}|))}{\log_2(n)}$$

Hence every limit of a sub-sequence of the sequence having general term $\frac{\log_2(\log_2(N_n(X)))}{\log_2(n)}$ is smaller than d. □

Proposition 2.37. *A subshift X which has positive entropy has entropy dimension equal to d.*

Proof. If $h(X) > 0$, there exists some n_0 such that for all $n \geq n_0$,

$$\frac{\log_2(N_n(X))}{n^d} \geq \frac{h(X)}{2} > 0.$$

As a consequence, for all these n,

$$N_n(X) \geq 2^{n^d h(X)/2},$$

and

$$\frac{\log_2(\log_2(N_n(X)))}{\log_2(n)} \geq d + \frac{h(X)}{2\log_2(n)} \to d.$$

Since the two entropy dimension are smaller than d, thanks to Proposition 2.36, they are equal to d. □

We state the following proposition in order to show that the entropy dimension is very different from the entropy. This type of properties has an impact on the techniques used in order to realize numbers as topological invariants of subshifts.

Proposition 2.38. *Let X and Z be two $\mathbb{Z}^d$-subshifts having entropy dimension. The subshift $X \times Z$ has entropy dimension:*

$$D_h(X \times Z) = \max(D_h(X), D_h(Z)).$$

Proof. We have the following for all $n \geq 1$:

$$N_n(X \times Z) = N_n(X) \times N_n(Z).$$

As a consequence,

$$\log_2(N_n(X \times Z)) = \log_2(N_n(X)) + \log_2(N_n(Z)).$$

Thus we have

$$\frac{\max(\log_2 \circ \log_2(N_n(X)), \log_2 \circ \log_2(N_n(Z)))}{\log_2(n)} \leq \frac{\log_2 \circ \log_2(N_n(X \times Z))}{\log_2(n)}$$

and

$$\frac{\log_2 \circ \log_2(N_n(X \times Z))}{\log_2(n)} \leq \frac{\log_2(2\log_2(\max(N_n(X), N_n(Z)))}{\log_2(n)}$$

From this follows that

$$\frac{\log_2 \circ \log_2(N_n(X \times Z))}{\log_2(n)} \leq \frac{\log_2(2) + \max(\log_2 \circ \log_2(N_n(X)), \log_2 \circ \log_2(N_n(Z)))}{\log_2(n)}.$$

As a consequence of the previous inequalities, we have, taking $n \to +\infty$,

$$D_h(X \times Z) = \max(D_h(X), D_h(Z)).$$

□

2.2.4.3 Effect of dynamical properties

Proposition 2.39. *1. A subshift X which has at least two elements and is K-block gluing verifies $h(X) > 0$.*

2. A minimal SFT has entropy dimensions smaller than $d-1$.

3. A minimal SFT has entropy zero: $h(X) = 0$.

Proof. 1. Let X be a non-trivial K-block gluing subshift, and $n \geq 1$.

(a) **Use of the block gluing property on sets of 1-blocks:**
Consider any sequence of 1-blocks in the language of X (there are at least two), $(a_{i_1,\dots,i_d})_{i_j \leq k}$ for some k. Then one can obtain, by applying successively the block gluing property on these 1-block, some $(k+(k-1)K)$-block p such that for all $\mathbf{n} \in \{0, k-1\}^d$ the restriction of p on

$$(K+n)\mathbf{i} + \mathbb{U}_n$$

is the 1-block $a_{n_1,\dots,n_d}$.

(b) **Consequence on the complexity sequence:**
Hence the number of $k1+(k-1)K$-block in the language of X is greater than 2^{k^d}. This means that

$$\frac{\log_2(N_{k+(k-1)K})}{(kn+(k-1)K)^d} \geq \frac{k^d}{(k+(k-1)K)^d},$$

which tends towards 1. As a consequence, taking the limit when $k \to +\infty$,

$$h(X) \geq \frac{1}{(K+1)^d} > 0.$$

2. The proof of the second assertion is stated and proved in Chapter 6 as Proposition 6.1.

3. A minimal SFT has entropy dimensions smaller than $d-1$. It means that there exists some n_0 such that for all $n \geq n_0$

$$\frac{\log_2(\log_2(N_n(X)))}{\log_2(n)} \leq d - 1/2.$$

This means that for all $n \geq n_0$,

$$N_n(X) \leq 2^{n^{d-1/2}},$$

and

$$\frac{\log_2(N_n(X))}{n^d} \leq \frac{1}{\sqrt{n}},$$

which means that $h(X) = 0$.

□

Chapter 3

Subshifts and computability

Sommaire

In this chapter, we present aspects of subshifts related to computability theory. First we present the Turing machine model and notions of computability for real numbers and $\mathbb{Z}^d$-subshifts. Using known results, we give some evidence of the effect of dynamical restrictions on computational aspects of these subshifts. Then we focus on growth-type invariants and give known characterizations of these invariants with computability conditions. The aim of the following chapters is to understand the effect of dynamical restrictions on the possible values of these growth-type invariants. The most difficult and interesting part in this problem is the construction of subshifts of finite type realizing the possible values of these invariants. In Section 3.4, the original part of this chapter, we provide the main principles and tools used in these constructions, abstracting the general scheme of construction of this text and the constructions in the literature. We describe a minimal version of the Robinson subshift, used as a structure in the constructions of the following chapters.

3.1 Computing machines

In this section, we present a model of computing machines. This model will be refined in the other chapters when needed.

3.1.1 Turing machines

Definition 3.1. *A* ***Turing machine*** $\mathcal{T}$ *is some tuple* $(\mathcal{Q}, \mathcal{A}, \delta, q_0, q_h, \#, \$)$*, where :*

- $\mathcal{Q}$ *is some finite set, called the* ***state set*** *of the machine,*
- $\mathcal{A}$ *is also a finite set, called the* ***alphabet*** *of the machine,*
- δ *is some function*
$$\delta : \mathcal{A} \times \mathcal{Q} \to \mathcal{A} \times \mathcal{Q} \times \{\leftarrow, \rightarrow, \uparrow\},$$
and is called the ***transition function****. We denote* $\delta(a,q) = (\delta_1(a,q), \delta_2(a,q), \delta_3(a,q))$ *for all* $(a,q) \in \mathcal{A} \times \mathcal{Q}$*.*
- q_0, q_h *are elements of* $\mathcal{Q}$ *called respectively the initialization state, and the halting state; the second one verifies*
$$(\delta(a, q_h))_3 = \uparrow$$
for all $a \in \mathcal{A}$*. The other states* $q \neq q_h$*,*
$$(\delta(a, q))_3 \neq \uparrow .$$
- $\#$ *is some element of* $\mathcal{A}$ *called the blank symbol,*
- *and* $\$$ *is another element of* $\mathcal{A}$ *called separating symbol.*

Remark 13. *The definition is slightly different from the usual definition, in which the set* $\{\leftarrow, \rightarrow, \uparrow\}$ *is replaced by* $\{\leftarrow, \rightarrow\}$*. In our context, these symbols correspond to the description of the machine trajectory when we will code it in a subshift. Moreover, in the constructions of the following chapters, the halting state may be replaced by a set of states, each one having a specific function.*

3.1.2 Dynamical process of the Turing machines

A machine defines a dynamical process (the moving head machine according to the vocabulary introduced by P. Kurka [Kur97]) on $\mathbb{N} \times \mathcal{Q} \times \mathcal{A}^{\mathbb{N}}$, which corresponds to a function $\mathcal{T}_{\mathbb{N}}$ such that :

- when $n \geq 1$ or ($n = 0$ and $\delta_3(q, a_0) = \rightarrow$) :
$$\mathcal{T}_{\mathbb{N}}(n, q, (a_k)_k) = (n + \epsilon(q, a_n), \delta_2(q, a_n), (a_0, ..., a_{n-1}, \delta_1(q, a_n), a_{n+1}, ...)),$$
where $\epsilon(q, a_n) = 1$ if $\delta_3(q, a_n) = \rightarrow$ and -1 if $\delta_3(q, a_n) = \leftarrow$, and $\epsilon(q, a_n) = 0$ if $\delta_3(q, a_n) = \uparrow$.
- When $\delta_(a_0, q) = \leftarrow$:
$$\mathcal{T}_{\mathbb{N}}(0, q, (a_k)_k) = (0, \delta_2(q, a_0), (\delta_1(q, a_0), a_1, ..., , a_{n+1}, ...)).$$

Remark 14. *In the following, the second case of this definition will be treated differently. Indeed, when embedding computation into subshifts of finite type we can use local rules so that the definition of this process takes into account the geometry of the structures supporting computations, as in 3.4.1.3.*

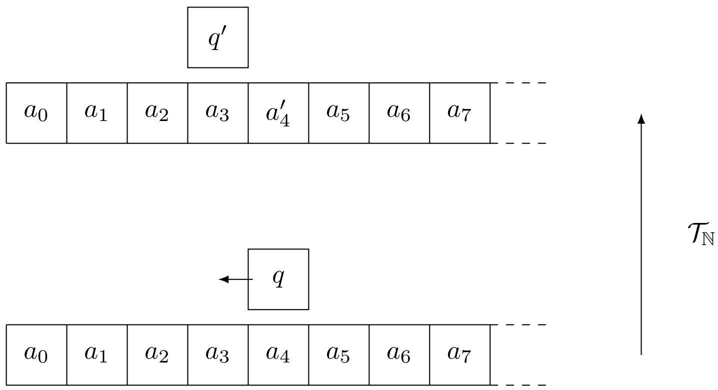


Figure 3.1: Illustration of the $\mathcal{T}_{\mathbb{N}}$ dynamical system. The arrows denote $\delta_3(q, a_4)$, $q' = \delta_2(q, a_4)$, and $a_4' = \delta_1(q, a_4)$.

We often represent the function $\mathcal{T}_{\mathbb{N}}$ as on Figure 3.1.

From this representation derive some words used to talk about a Turing machine : the first coordinate of an element of $\mathbb{N} \times \mathcal{Q} \times \mathcal{A}^{\mathbb{N}}$ is thought as a position in $\mathbb{N}$ of a head. The second coordinate is the state of this head. The third coordinate, in $\mathcal{A}^{\mathbb{N}}$, is thought as an infinite word written on a **tape**. During the process, the head moves on the tape. At each step of its movement, it reads the letter which is under it. According to this letter and its current state, it changes its state, and erases the letter. Then, it writes another one on this position and moves a step forward in the direction of the arrow given by the transition function.

When considering an orbit $\left\{\mathcal{T}_{\mathbb{N}}^k(x) : k \in \mathbb{N}\right\}$, $x \in \mathbb{N}\times\mathcal{Q}\times\mathcal{A}^{\mathbb{N}}$, of this system - meaning the trajectory of x under the action of the machine - we often call the state of the head (resp. position and infinite word written) in the configuration x the **initial state** (resp. position, tape).

Remark 15. *This definition describes a process involving a unique head. However, in the following chapters we will have to use a dynamical process derived from the Turing machine and which involves multiple heads. At this point, we will have to adapt the process to take into account the interactions between heads.*

When there is a unique machine head, an infinite word which consists in the concatenation of a finite set of finite words on $\mathcal{A}\backslash\{\#, \$\}$ separated by a \$ symbol, and followed by an infinity of # symbols (denoted $\#^\infty$) and written on the initial tape is called an **input**. When such an infinite word is written on the tape when the unique machine head enters in halting state, we call this word the **output** of the machine.

The computing machines often have alphabet $\mathcal{A} = \{0, 1, \#, \$\}$ and the inputs and outputs are finite sequences of binary representations of some integers. Sometimes the machine uses the unary representation of integers.

A machine can use local variables using multiple tapes : in this case the tape is some $\mathbb{N}^k$ for some $k \geq 1$ for each variable it uses, and each position is written with an element of $\mathcal{A}_1 \times ... \times \mathcal{A}_k$, where the $\mathcal{A}_i$ are finite alphabets. We will use this in the following without mentioning.

A machine also defines some dynamical process on $\mathbb{Z} \times \mathcal{Q} \times \mathcal{A}^{\mathbb{Z}}$ which corresponds to a function $\mathcal{T}_{\mathbb{Z}}$ such that for all $n \in \mathbb{Z}$:

$$\mathcal{T}_{\mathbb{Z}}(n, q, (a_k)_k) = (n + \epsilon(q, a_n), \delta_2(q, a_n), (..., a_{n-1}, \delta_1(q, a_n), a_{n+1}, ...)),$$

where $\epsilon(q, a_n) = 1$ if $\delta_3(q, a_n) = \rightarrow$ and -1 if $\delta_3(q, a_n) = \leftarrow$, and $\epsilon(q, a_n) = 0$ when $\delta_3(q, a_n) = \uparrow$.

In this case the pathological case of head movement disappears.

Remark 16. *It is known that for all the finite sequences of arithmetical operations using local variables (an algorithm) there is a Turing machine that executes this sequence of operations on the input and then halts. The output of the algorithm is written on the tape at halting time. We use this known fact in the following of this text and often make the confusion between machines and algorithms.*

3.1.3 Universality

We will use as well the following well-known properties of the Turing machine model. More details can be found in [Rog87].

Theorem 3.2 (Turing)**.** *There exists a Turing machine $\mathcal{T}_{univ}$ that:*

1. *given as input the binary representations of some integers n, m, t halts and outputs a word on $\{0, 1\}$,*
2. *and for any turing machine $\mathcal{T}$, there exists some n such that on input n, m, t the machine $\mathcal{T}_{univ}$ outputs the word written on the tape of the machine $\mathcal{T}$ on input m and at after t computation steps.*

The machine $\mathcal{T}_{univ}$ is called a universal Turing machine.

Theorem 3.3 (Turing)**.** *There is no Turing machine which takes as input some n and outputs* 0 *if the Turing machine $\mathcal{T}_n$ halts on input* 0*, and* 1 *if not (we would say that this Turing machine would **decide** the halt of the Turing machines).*

Remark 17. *This is often refered as the undecidability of the halting problem.*

3.1.4 Space-time diagrams

The set of space-time diagrams of $\mathcal{T}_{\mathbb{N}}$, for a Turing machine $\mathcal{T}$, is the subset of $(\mathcal{A} \cup \mathcal{A} \times \mathcal{Q})^{\mathbb{N}^2}$ such that in all the configurations of this set, all the rows of $\mathbb{N}^2$ are written with a representation of an element of $\mathbb{N} \times \mathcal{Q} \times \mathcal{A}^{\mathbb{N}}$, $(n, q, \overline{a})$, where the machine head is in position n. We represent this element by a sequence whose kth term is a_k for all $k \neq n$, and the nth one is (a_n, q).

For two adjacent rows the above one is the image by $\mathcal{T}_{\mathbb{N}}$ of the row below, and the first row is an initialization one. This set represents all the possible trajectories of a machine. See Figure 3.2 for an illustration.

We will see in the next sections that we can embed arbitrary large parts of this space-time diagram in subshifts of finite type.

3.2 Computability notions

In this section we present some notions that are derived from the Turing machine model. Informally, considering a class of objects that possesses a canonical way to describe the objects, we say that an object is computable when its description is the output of a Turing machine.

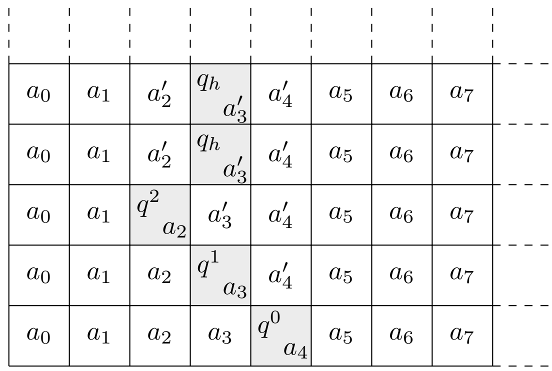


Figure 3.2: A part of a space-time diagram of a Turing machine.

3.2.1 Integer functions

Definition 3.4. *1. A* ***computable function*** *$f : \mathbb{N}^m \to \mathbb{N}^n$ is a function such that there exists a Turing machine which, on the input which consists in the binary representation of a m-uple integers, outputs the binary representation of its image by f.*

2. As well we say that a sequence in $\mathcal{U}^{\mathbb{N}^m}$, where $\mathcal{U}$ is some finite set, is computable, when there exists a Turing machine which on input n, outputs the nth element of the sequence.

3.2.2 Arithmetical hierarchy of real numbers

The following notion relies on the description of the real numbers as limits of sequences of rational numbers.

Let $\overline{\Theta}_j$, with $j \geq 1$, the operator on $\mathbb{Q}^{\mathbb{N}^j}$ that to a sequence $(\epsilon_{n_1,...,n_j})_{(n_1,...,n_j)\in\mathbb{N}^j}$ associate the sequence

$$(\sup_{n_1} \epsilon_{n_1,...,n_j})_{(n_2,...,n_j)\in\mathbb{N}^j}.$$

As well, the operator $\underline{\Theta}_j$ associates to the sequence

$$(\inf_{n_1} \epsilon_{n_1,...,n_j})_{(n_2,...,n_j)\in\mathbb{N}^j}.$$

Definition 3.5 ([Wei13])**.**

- *A real number x is called Σ_j-computable if there exists a Turing machine that, taking as input some j-uple of integers $(n_1, ..., n_j)$, stops and outputs the representation of a rational number $r_{n_1,...,n_j}$ such that*

$$x = \left(\overline{\Theta}_j \circ \underline{\Theta}_{j-1} \circ ...\right)(r),$$

  where r is the sequence of the numbers $r_{n_1,...,n_j}$.

- *A real number x is called Π_j-computable if there exists a Turing machine that, taking as input some j-uple of integers $(n_1, ..., n_j)$, halts and outputs the representation of a rational number $r_{n_1,...,n_j}$ such that*

$$x = \left(\underline{\Theta}_j \circ \overline{\Theta}_{j-1} \circ ...\right)(r),$$

  where r is the sequence of the numbers $r_{n_1,...,n_j}$.

- *A real number is said Δ_j-computable when it is Σ_j-computable and Π_j-computable. A Σ_1 computable number (resp. Π_1-computable) is also called lower (resp. upper) semi-computable. When it is Δ_1-computable, we say that it is computable.*

The real numbers are represented by elements of $\mathbb{Z} \times \{0,1\}^{\mathbb{N}}$ (the integer represents the integer part, and the sequence of $0,1$ the base two representation of the fractional part). We characterize the computable numbers as the set of numbers having computable representation:

Proposition 3.6. *Let x be a real number. The following properties are equivalent:*

1. *the real number x is computable.*
2. *there is a Turing machine that taking as input some integer $n \geq 1$, outputs some $\epsilon_n \in \{0,1\}$. The sequence $(\epsilon_n)_n$ verifies:*
$$x - \lfloor x \rfloor = \sum_{n=1}^{+\infty} \frac{\epsilon_n}{2^n}.$$
3. *there exists a Turing machine that taking as input some integer n, outputs some $r_n \in \mathbb{Q}$ such that $|x - r_n| \leq 2^{-n}$.*

Proof.

- 1. $\Rightarrow$ 3.

  Assume that x is computable.

  1. **Formulation of the definition:** This means that there exist two algorithms that, given as input some integer n, output respectively some rational numbers l_n and r_n such that $x = \sup_n l_n = \inf_n r_n$.
  2. **The speed of convergence of two adjacent sequences can be computed:**
     Since the two sequences converge towards x, this means that for all k, there exists some n_k minimal such that $|l_{n_k} - r_{n_k}| < 2^{-k}$. The function $k \mapsto n_k$ is computable: consider the algorithm that given as input some k:
     (a) computes successively the numbers $l_n - r_n$, $n \geq 1$,
     (b) computes each time the difference $|l_n - r_n|$ and checks if this difference is less than 2^{-k},
     (c) and when it is halts and outputs n_k.
  3. **Computability of the number x:**
     Hence there is an algorithm that given as input some k outputs l_{n_k}. Since $x \in [\![l_{n_k}, r_{n_k}]\!]$, and $|l_{n_k} - r_{n_k}| \leq 2^{-k}$, this means that $|l_{n_k} - x| \leq 2^{-k}$. As a consequence x is computable.

- 3. $\Rightarrow$ 2. Assume that there exists a Turing machine that taking as input some integer n, outputs some $r_n \in \mathbb{Q}$ such that $|x - r_n| \leq 2^{-n}$.

  1. **The sequence can be taken non-decreasing or non-increasing:**
     We can assume that $r_n \leq x$ by adding -2^{-n} to r_n when $x < r_n$. In this case, this does not change the inequality $|x - r_n| \leq 2^{-n}$. Moreover, we can consider that the sequence $(r_n)_n$ ca be considered non-decreasing by replacing r_n by $\max_{k \leq n} r_k$. Indeed, this sequence is still computable and converges towards x.
  2. **Computing a representation of x:**
     Let us define the sequences $(d_n)_n$ and $(\epsilon_n)_{n\geq 1}$ such that for all $n \geq 1$, $\epsilon_n \in \{0,1\}$ and
$$d_n = \sum_{k=1}^{n} \frac{\epsilon_k}{2^k}.$$

We define ϵ_1 maximal such that $\epsilon_1/2 \leq r_1$. As a consequence $|d_1 - r_1| \leq 2^{-1}$, and thus $|d_1 - x| \leq 2.2^{-1}$. For all $n \geq 2$, we defined ϵ_n to be maximal such that

$$d_{n-1} + \frac{\epsilon_n}{2^n} \leq r_n.$$

As a consequence, $|d_n - r_n| \leq 2^{-n}$ and thus $|d_n - x| \leq 2.2^{-n}$.
As a consequence,

$$x = \sum_{k=1}^{+\infty} \frac{\epsilon_k}{2^k}.$$

From the fact that $(r_n)_n$ is computable, $(\epsilon_n)_n$ is computable.

- 2. $\Rightarrow$ 1. Assume that there is an algorithm that given as input an integer $n \geq 1$, outputs the nth bit of a base two decomposition of the fractional part of x, and on input 0 outputs the integer part of x. With the notations of the previous item, let

$$r_n = m + \sum_{k \leq n} \epsilon_k 2^{-k},$$

and

$$l_n = m + \sum_{k \leq n} \epsilon_k 2^{-k} + \sum_{k \geq n+1} 2^{-k},$$

where m is the integer part of x. Then the functions $n \mapsto r_n$ and $n \mapsto l_n$ are computable, and $x = \sup_n r_n = \inf_n l_n$, hence x is computable.

□

Corollary 3.7. *For all x, z real numbers, if x is computable and z is a Π_1 (resp Σ_1) computable number, then $z + x$ is Π_1 (resp. Σ_1) computable number.*

The following theorem says that the hierarchy of real numbers does not collapse:

Theorem 3.8 ([Wei13]). *For all integer $n \geq 1$, we have that $\Delta_n \subset \Sigma_n$, $\Delta_n \subset \Pi_n$, $\Sigma_n \subset \Delta_{n+1}$, and $\Pi_n \subset \Delta_{n+1}$, and all these inclusions are strict.*

Example 3.9. • ***The number e is a computable number:***

Indeed, for all $n \geq 0$

$$\sum_{k=n+2}^{+\infty} \frac{1}{k!} \leq \sum_{k=n+2}^{+\infty} \frac{1}{2^k} = \frac{1}{2^{n+1}}.$$

Moreover, there exists some algorithm that, given an integer n as input, outputs the rational number

$$r_n = \sum_{k=0}^{n+1} \frac{1}{k!}.$$

Since we have

$$|e - r_n| = \sum_{k+2}^{+\infty} \frac{1}{k!} \leq 2^{-n-1} \leq 2^{-n},$$

this means that the number e is computable.

- ***An example of strictly Σ_1-computable number:***
  Let u be the number defined as
  $$u = \sum_{k \in \mathcal{H}} \frac{1}{2^k},$$
  where $\mathcal{H}$ is the set of integers k such that the kth Turing machine $\mathcal{T}_n$ halts. This number is not computable. Indeed, if we assume the opposite, it implies that there exists an algorithm that given as input some k, outputs the kth bit of the base two decomposition of this number. Since this bit is given by 0 if the kth machine does not halt, and 1 if it does, then this algorithm can decide if a Turing machine halts or not. This is impossible.

  However, this number is Σ_1-computable. Indeed, there is an algorithm that given as input some integer n, computes the rational number
  $$r_n = \sum_{k=0}^{n} \epsilon_{k,n} \frac{1}{2^k},$$
  where $\epsilon_{k,n} = 1$ if the kth Turing machine halts before n steps of computation and 0 if not. We have that
  $$u = \sup_n r_n.$$

Definition 3.10. *Let $I, J \subseteq \mathbb{R}$ intervals, and $f : I \to J$ a function. We say that the function f is computable when there exists some computable function $\varphi : \mathbb{N} \to \mathbb{N}$ and an algorithm that given as input the integer part of some $x \in I$ and the $\varphi(n)$ first bits of a base two decomposition of the fractional part of x, computes the n first bits of the base two decomposition of $f(x)$.*

Proposition 3.11. *A function $I \to J$ is computable if and only if there exists some computable function $\varphi : \mathbb{N} \to \mathbb{N}$ and an algorithm that given as input the integer part of x and $\varphi(n)$ first bits of the base two decomposition of the fractional part of x outputs some rational number r_n such that $|f(x) - r_n| \leq 2^{-n}$.*

Proof. The proof is similar to the one of Lemma 3.6. It relies on the fact that being able with an algorithm to give an approximation of a real number x by some integer r_n such that $|x - r_n| \leq 2^{-n-2}$ implies being able to give the n first bits of a base two decomposition of the fractional part of x. Moreover Moreover, being able to give these bits implies being able to approximate x by some integer r_n such that $|x - r_n| \leq 2^{-n}$. □

Example 3.12 ([Wei13])**.** *The functions exp* $: \mathbb{R} \to \mathbb{R}$ *and* $\log_2 :]0, +\infty[\to \mathbb{R}$ *are computable.*

3.2.3 Effective subshifts

The notion of computability on subshifts relies on the description by a forbidden language via an encoding, fixed in the following, of the set of patterns on $\mathbb{Z}^d$ and on an alphabet $\mathcal{A}$. This is an onto computable map from the set of finite patterns on $\mathbb{N}$ and $\mathcal{A}$ to this set. A finite pattern on $\mathbb{N}$ and $\mathcal{A}$ **code for** a pattern on $\mathbb{Z}^d$ and $\mathcal{A}$.

Definition 3.13. *Let X be a $\mathbb{Z}^d$-subshift on alphabet $\mathcal{A}$. It is said to be* ***effective*** *when there exists some language $\mathcal{F}$ such that:*

1. *there exists a Turing machine which on input n outputs a code for some pattern p_n such that $\mathcal{F} = \{p_n : n \in \mathbb{N}\}$*
2. *and X is equal to $X_{\mathcal{F}}$.*

Remark 18. *In particular, since subshifts of finite type are obtained by forbidding a finite set of patterns, these subshifts are effective.*

3.3 Obstructions to computation

In this section we prove known facts about the computability of the entropy and entropy dimension of effective $\mathbb{Z}^d$-subshifts. We first prove a general obstruction, meaning a computability condition verified by the invariants of all effective subshifts.

3.3.1 General obstruction

The following theorem gives computability obstructions for the growth-type invariants studied in this text: entropy and entropy dimension. This is a generalization to effective $\mathbb{Z}^d$ subshifts of the same result for SFTs, proved in [HM10] and [Mey11].

Theorem 3.14 ([HM10])**.** *The entropy of an effective $\mathbb{Z}^d$-subshift is a Π_1-computable number.*

Proof.

1. **Formulation of the hypothesis:**

   Let X be an effective $\mathbb{Z}^d$-subshift. There exists a Turing machine that given as input some integer n, outputs some p_n such that $X = X_{\mathcal{F}}$ for $\mathcal{F} = \{p_n : n \in \mathbb{N}\}$.

2. **Globally admissible and partially locally admissible patterns:**

   Let (n, m, l) be a triple of integers such that $l \geq n$. Denote $N_{n,m,l}(X)$ the number of n-blocks appearing at the center of some l-block which does not have any of the $p_0, ...p_m$ as a sub-pattern - meaning some locally admissible pattern in the subshift $X_{\mathcal{F}_m}$, where $\mathcal{F}_m = \{p_0, ..., p_m\}$.

   For all n,
   $$N_n(X) = \inf_m \inf_{l \geq n} N_{n,m,l}(X).$$
   We prove this equality in the following points.

3. **Upper bound:**

   Indeed, for all such (n, m, l), since any globally admissible n-block p in X appears in some configuration x of this subshift, by extending this pattern into some l-block in this configuration, we get that p appears at the center of some l-block where none of the patterns p_i, $i \geq 0$ appear. In particular the patterns $p_0, ..., p_m$. As a consequence, $N_n(X) \leq N_{n,m,l}(X)$, and
   $$N_n(X) \leq \inf_m \inf_{l \geq n} N_{n,m,l}(X).$$

4. **Equality:**

   Assume that this inequality is strict:
   $$N_n(X) < \inf_m \inf_{l \geq n} N_{n,m,l}(X).$$
   This means that there exists some p which is not globally admissible in X and so that for all (n, m, l) such that $l \geq 0$, there exists some admissible pattern $q_{n,m,l}$ on $[\![-l, n+l]\!]^d$ which is locally admissible in $X_{\mathcal{F}_m}$ and whose restriction on $[\![1, n]\!]^d$ is p.

   There exists some sub-sequence $(q^{(1)}_{n,l,l})_{l \geq n}$ of the sequence $(q_{n,l,l})_{l \geq n}$ whose elements all agree on $[\![-1, n+1]\!]^d$. Let us assume that a sequence $(q^{(k)}_{n,l,l})_{l \geq n}$ is defined. There exists some sub-sequence of this sequence, that we denote $(q^{(k+1)}_{n,l,l})_{l \geq n}$ whose elements all agree on $[\![-(k+1), n+(k+1)]\!]^d$.

   Let us denote x the configuration which agree with $(q^{(k)}_{n,l,l})_{l \geq n}$ on $[\![-k, n+k]\!]^d$ for all k. This means that none of the pattern of $\mathcal{F}$ appear in x. Hence x is an element of X, and p appears in this configuration: this is a contradiction.

5. **Computability of the entropy:**

   As a consequence
$$h(X) = \inf_n \frac{\log_2\left(\inf_m \inf_{l \geq n} N_{n,m,l}(X)\right)}{n^d} = \inf_{n,m} \inf_{l \geq n} \frac{\log_2(N_{n,m,l}(X))}{n^d},$$
   by continuity and increase of the logarithm function.

   Since the function $(n, m, l) \mapsto N_{n,m,l}(X)$ is computable, $h(X)$ is a Π_1-computable number. Indeed, the algorithm to compute this function is to check for all l blocks in the full shift if some p_i, $i \leq m$ appears in this block, and count the n-blocks appearing at the center of the patterns in which none of the patterns p_i, $i \leq m$ appears.

□

Theorem 3.15 ([Mey11])**.** *For all $d \geq 2$ and X an effective $\mathbb{Z}^d$-subshift:*

1. *The upper entropy dimension of X is Π_3-computable.*
2. *The lower entropy dimension of X is Σ_2-computable*
3. *If X has an entropy dimension, this number is Δ_2-computable.*

Proof. Let $X = X_{\mathcal{F}}$ an effective subshift. We re-use the formula used in the proof of Theorem 3.14: for all n,
$$N_n(X) = \inf_m \inf_{l \geq n} N_{n,m,l}(X).$$

The upper entropy dimension of X is
$$\overline{D_h}(X) = \limsup_n \frac{\log_2(\log_2(N_n(X)))}{\log_2(n)} = \inf_n \sup_{k \geq n} \inf_{m,l \geq k} \frac{\log_2(\log_2(N_{k,m,l}(X)))}{\log_2(k)}.$$

As a consequence, since the function $k, m, l \mapsto N_{k,m,l}(X)$ is computable, this number is Π_3-computable.

The lower entropy dimension of X is
$$\underline{D_h}(X) = \liminf_n \frac{\log_2(\log_2(N_n(X)))}{\log_2(n)} = \sup_n \inf_{k \geq n} \inf_{m,l \geq k} \frac{\log_2(\log_2(N_{k,m,l}(X)))}{\log_2(k)}.$$

For the same reason, this is a Σ_2-computable number.

If the lower and upper entropy dimensions are equal, the number $D_h(X)$ is a Δ_2-computable number, since it is expressed as the limit of a computable sequence. □

When the dimension d is one, the obstruction is stronger: the entropy becomes computable. This relies on the following theorem.

Theorem 3.16 ([LM95])**.** *The numbers that are entropy of a $\mathbb{Z}$-SFT are the $\log(\lambda)$, where λ is a Perron number, meaning the maximal eigenvalue of matrix with non-negative entries.*

Corollary 3.17. *There is an algorithm which given as input a finite set of forbidden words $\mathcal{F}$ and an integer n outputs some rational number r_n such that $|h(X_{\mathcal{F}}) - r_n| \leq 2^{-n}$. We say that the entropy of $\mathbb{Z}$-SFTs is uniformly computable. In particular, the entropy of a $\mathbb{Z}$-SFT is computable.*

3.3.2 Realization in the non-constrained case

For multidimensional subshifts, the possible entropies and entropy dimensions are in fact characterized by the obstruction, as stated by the following theorems. The interest of the texts [HM10] and [Mey11], lies in the constructive methods developed by the authors, using an implementation of Turing machines into subshifts of finite type, which control random bits generating the entropy. The constructions presented in this text are mostly based on these methods.

Theorem 3.18 ([HM10]). *For all $d \geq 2$, the possible entropies of $\mathbb{Z}^d$-SFT are the Π_1-computable numbers.*

Theorem 3.19 ([Mey11]). *For all $d \geq 2$, the numbers that are entropy dimension of a $\mathbb{Z}^d$-SFT are the Δ_2-computable numbers.*

3.3.3 Known effects of dynamical properties

The properties of minimality and block gluing have an effect on computability properties of effective subshifts.

Proposition 3.20 (Folklore). *The language of a minimal non empty effective subshift is decidable.*

Proof. Let $X = X_{\mathcal{F}}$ be a non empty minimal subshift, with $\mathcal{F} = \{p_n : n \in \mathbb{N}\}$ for $n \mapsto p_n$ a computable function.

1. **Globally admissible patterns and partially locally admissible ones under minimality constraint:**

   A pattern p is globally admissible if and only if there exists some n such that any pattern q on $[\![-n, n]\!]^d$, which has none of the patterns p_k, $k \leq n$ as sub-pattern, has p as sub-pattern.

   Indeed, assume that p is globally admissible, and this condition is false for all n, and denote q_n an exception to this condition.

   One can complete q_n into a configuration x_n of the full-shift. By compactness, there exists a configuration x of the full-shift such that for all n, when l is great enough, x agrees with all the x_m, $m \geq l$ on $[\![-n, n]\!]^d$.

   This means that none of the elements of $\mathcal{F}$ appear in x. Thus x this is an element of X which has not p as sub-pattern: this contradicts the minimality.

   Moreover, a pattern which verifies the above condition is globally admissible since any configuration contains this pattern.

2. Recall (Lemma 2.11) that a pattern p is not globally admissible if and only if there exists some n such that all the n-blocks in which none of the pattern p_k, $k \leq n$ appears, do not have p as sub-pattern.

3. **Decidability:**

   The following algorithm decides if a pattern p is admissible in X or not:

   (a) The algorithm successively computes the patterns $p_0, ..., p_n$ and checks, for any pattern in $\mathcal{A}^{[\![-n,n]\!]^d}$ having none of the patterns $p_0, ...p_n$ as sub-pattern, if p appears in this pattern.

   (b) If for some n, this condition is true, then outputs 1. If for some n, all these patterns do not contain p, then output 0.

This algorithm stops on any input. Indeed, there exists n such that one of the two conditions of the second point in this description is verified.

$\square$

Theorem 3.21 ([PS15]). *If X is K-block gluing for some $K > 0$, then there exists some $C > 0$ and a Turing machine which, on input n, outputs an approximation of the entropy of X up to precision 2^{-n}, in less than $O(2^{Cn^d})$ steps.*

Reciprocally, R. Pavlov and M. Schraudner [PS15] proved that a sub-class of the computable numbers is realizable as entropy of K-block gluing $\mathbb{Z}^3$-SFT for some $K > 0$. This result is not a characterization: the exact class of numbers that are the entropy of a K-block gluing SFT is still not known.

These results lead to the natural question of the limit constraints (in particular dynamical ones) for which the entropy of a $\mathbb{Z}^d$-SFT becomes computable. What kind of dynamical restrictions allow universal computation, involved in the proofs of M. Hochman and T. Meyerovitch theorems ? In this context, how one has to adapt the computation model to the constraints ?

The aim of this text is to present characterizations of growth type invariants of $\mathbb{Z}^d$-SFT under constraints of minimality and block gluing with gap function. In the next section, we present how are organized our constructions, and the difficulties encountered while programming SFT under these constraints.

3.4 Programming subshifts of finite type under dynamical constraints

3.4.1 General organization of the constructions

In this section, we explain in an informal way the general organization of our constructions. The subshifts that we construct are divided into multiple layers, each one corresponding to an intuitive function. Each layer has specific kind of rules, and interaction rules between the layers.

The general decomposition into layers is as follows:

1. A structure layer, which exhibits some hierarchical structures used to transport information.

2. A basis or information layer that contains some data localized according to the structure.

3. One or more synchronization layers, which allow the synchronization of the data amongst the levels of the hierarchy.

4. A computation layer, that supports the computation of particular Turing machines.

In a construction we present first the structure layer and then add successively other layers with internal rules (meaning that these forbidden patterns are not depending on the alphabets of the other layers) and interaction rules (meaning that they are not internal rules). Each time we add some $(k+1)$th layer, the alphabet $\mathcal{A}_1 \times ... \times \mathcal{A}_k$ is changed to $\mathcal{A}_1 \times ... \times \mathcal{A}_k \times \mathcal{A}_{k+1}$, and the set of rules is changed by adding the rules described for the $(k+1)$th layer.

3.4.1.1 Structure and information storage

The structure layer allows the other behaviors programmed in the subshift to be localized. This means that for all the other layers, there will be an interaction rule with the structure layer called **localization rule**. This rule specifies which symbols in the alphabet $\mathcal{A}_i$ of the ith layer, $i > 1$ can coexist with a symbol in $\mathcal{A}_1$, the alphabet of the structure layer, on a position.

Example:

We present here an example of a $\mathbb{Z}^2$-structure layer, denoted T_1, which will be followed in this section by an example of additional information layer and synchronization layer.

13	23	33	03	13	23	33	03	13	23
12	22	32	02	12	22	32	02	12	22
11	21	31	01	11	21	31	01	11	21
10	20	30	00	10	20	30	00	10	20
13	23	33	03	13	23	33	03	13	23
12	22	32	02	12	22	32	02	12	22
11	21	31	01	11	21	31	01	11	21
10	20	30	00	10	20	30	00	10	20
13	23	33	03	13	23	33	03	13	23
12	22	32	02	12	22	32	02	12	22

Figure 3.3: Example of a configuration in T_1.

Let $N \geq 2$ some integer. The alphabet $\mathcal{A}_1$ of T_1 is the set of symbols

$$\boxed{0j}, \boxed{i0}, \boxed{Nj}, \boxed{iN},$$

with $i, j \in \mathbb{Z}/(N+1)\mathbb{Z}$, and the symbols

$$\boxed{ij},$$

with $i, j \in \mathbb{Z}/(N+1)\mathbb{Z}\backslash\{\overline{0}, \overline{N}\}$. The rules are the following ones: on the right of a symbol marked with ij, the symbol is marked with $(i+1)j$, and on the top of this same symbol, the symbol is marked with $i(j+1)$. Figure 3.3 shows an example of configuration in this subshift, for $N = 3$.

The symbols of this first layer are often partially represented, in order to put forward some structures exhibited by the configurations of this layer.

One can superimpose on this layer a second layer that we call **information layer** on alphabet $\mathcal{A}_2 = \{0, 1, \square\}$, hence forming a subshift T_2 on alphabet $\mathcal{A}_1 \times \mathcal{A}_2$, with the additional rule that the symbols $0, 1$ can appear only on the positions $\mathbf{z}$ of $\mathbb{Z}^2$ where the structure layer symbol is $\blacksquare$ with the same symbol on positions $\mathbf{z} + \{(0,1), (0,-1), (1,0), (-1,0)\}$. Figure 3.4 shows a possible configuration of the information layer over the previous configuration of the structure layer.

We call this layer information layer because it contains some bits of information (these bits can be chosen freely) in the set $\{0, 1\}$, localized on the centers of the gray crosses. For x a configuration of T_2, let us denote $\mathbb{I}_x$ the set of positions that support some bit of information. In this example, all the set $\mathbb{I}_x$ are some $\mathbf{u} + N\mathbb{Z}^2$, with $\mathbf{u} \in \{0, ..., N-1\}^2$.

Remark 19. *The symbol $\square$ is a symbol used to signify that there is no information. We refer to this symbol as the **blank symbol** in the following of this text.*

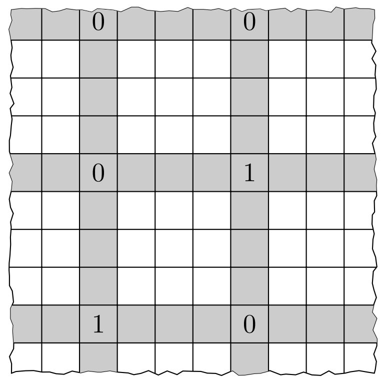


Figure 3.4: Example of a configuration of T_2, whose projection on the structure layer is given by Figure 3.3. Symbols of the structure layer are partially represented.

3.4.1.2 Signals and synchronization

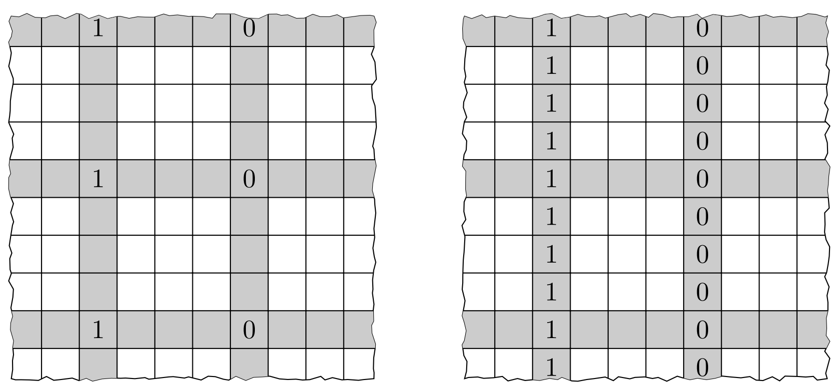


Figure 3.5: Illustration of the rules of the synchronization layer. As previously, the structure symbols are partially represented. The picture on the left represents a possible projection on the information and structure layers of a configuration in T_3. The one on the right represents a possible projection on the structure and synchronization layer of the same configuration.

A **synchronization layer** is generally used for synchronizing the bits of the information layer on subsets of $\mathbb{I}_x$ in any configuration x of the subshift: this means that, if $\mathbb{I}_x = \bigcup_{i\in\mathbb{Z}} \mathbb{I}_x^i$ is some decomposition of $\mathbb{I}_x$ for all x in the product of the structure layer and the information layer, the bits of the information layer on all the positions of some $\mathbb{I}_x^i$ are equal.

The synchronization is done using **signals**. These signals are rules telling that for two adjacent positions of $\mathbb{Z}^d$ having the same symbol in the structure layer, the pair of symbols in the synchronization layer verifies some relation. This relation depends on the symbols of the structure layer in a fixed neighborhood. A **synchronization rule** is used to tell that on the positions where an information is placed, the symbol in the information layer is equal to the symbol in the synchronization layer.

Remark 20. *The notion of signal was already used in an intuitive way in the original work of R. Berger [Ber66]. Here we attempt to give a formal definition of a signal.*

In our example, the synchronization layer could be the following one. The alphabet is

$$\mathcal{A}_3 = \{0,1\}^2 \cup \{\square\},$$

and the rules are the following ones:

- **Localization rule:** the non-blank symbols are superimposed on the positions $\mathbf{z}$ whose symbol in the structure layer is $\square$, and either the two symbols on $\mathbf{z}+(1,0)$ and $\mathbf{z}+(-1,0)$ are not $\square$, or the four symbols on positions in $\mathbf{z}+\{(1,0),(-1,0),(0,1),(0,-1)\}$ are $\square$.

- **Signal:** on two adjacent positions corresponding to the description of the localization rule, the symbols in this layer are equal.

- **Synchronization:** on positions where there is some information placed, the bit of the information layer is equal to the bit of the synchronization layer.

The total subshift, denoted T_3, is now on alphabet $\mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3$, and the projection of a typical configuration in this subshift is as on Figure 3.5 , on the left side, while the corresponding picture in the synchronization layer is on the right side.

In this example, the sets $\mathbb{I}_x^i$, $i \in \mathbb{Z}$, are the sets

$$\mathbb{I}_x^i = (k_x + Ni, 0) + \mathbb{Z}(0,1),$$

for some $k_x \in \{0, ..., N-1\}$.

Remark 21. *One can distinguish two modes of communication. The first one,* ***wire transport of information****, corresponds to the situation where the transmission rules is restrictive with respect to the possible directions for the signal propagation, as in the example. The second one is* ***diffusive transport****, which corresponds to a transmission rule that does not restrict the possible directions for propagation, as on Figure 3.6.*

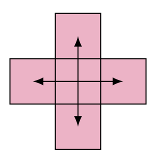


Figure 3.6: Illustration of the diffusive transport of information

3.4.1.3 Initialized computation

Let $(\mathcal{Q}, \mathcal{A}, \delta)$ be a Turing machine.

One can add another layer, called **machine layer**. The purpose of this layer is to execute some computations on some data supported by the information layer.

It has *alphabet* $\mathcal{A}_4 = \mathcal{A} \times \{\leftarrow, \rightarrow\} \cup \mathcal{A} \times \mathcal{Q} \times \{\leftrightarrow\} \cup \{\square\}$.

The *rules* are the following ones:

1. **Localization:** the non-blank symbols are superimposed on and only on positions that are colored $\square$ in the structure layer. A connected set of positions $\mathbb{A}_x$ that are colored $\square$ in a configuration x is called a **computation area**.

2. **Rules for the machine process when there is no entering head:** without consideration on the arrows, on a 3×2 pattern having only non-blank symbols, if the bottom row of the pattern contains no head, then the symbols on positions $(1, 2)$ and $(2, 2)$ - where the position $(1, 1)$ is the leftmost and bottommost one - are equal. For instance:

u	v	w
u	v	w

,

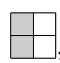
or when a head enters:

u	v	(w, q)
u	v	w

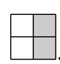
3. - **Rules for the machine process when there is an entering head:** When there is a machine head in the bottom row in state q and with letter a: if the head is on the leftmost position and $\delta_3(a, q) = \leftarrow$, or is on the rightmost position and $\delta_3(a, q) = \rightarrow$, then the top row contains only letters and these ones are copied from the bottom row.
   - If the head is in position $(1, 2)$ and $\delta_3(a, q) = \leftarrow$, then in the top row the head is in position $(2, 1)$ and in state $\delta_1(a, q)$ and the letter in position $(2, 2)$ is $\delta_2(a, q)$. If the head is in position $(1, 2)$ and $\delta_3(a, q) = \rightarrow$, then in the top row the head is in position $(2, 3)$ and in state $\delta_1(a, q)$ and the letter in position $(2, 2)$ is $\delta_2(a, q)$. The other letter is equal to the corresponding one in the bottom row. For instance:

u	(v, q_2)	x
u	v	(w, q_1)

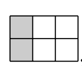
4. **On the sides of an area:** on a pattern

,

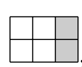
if the symbol on the $(2, 1)$ position is $(a, q, \leftrightarrow)$ such that $\delta(a, q)_3 = \leftarrow$, then the symbol on position $(2, 2)$ has no head, meaning it is in $\mathcal{A} \times \{\rightarrow, \leftarrow\}$. A similar rule is true on a pattern

.

5. On a pattern

,

and similarly on

,

if the $(2, 1)$ position symbol is not some $(a, q, \leftrightarrow)$ such that $\delta(a, q)_3 = \uparrow$ and the $(3, 1)$ position symbol is not some $(a, q, \leftrightarrow)$ such that $\delta(a, q)_3 = \leftarrow$, then the position $(2, 2)$ has not head symbol.

6. **Uniqueness of the machine head in an area:** On two horizontally adjacent positions having non-blank symbol in this layer, the directions of the arrows have to match.

7. **Initialization rules:** On the bottom row of a computation area, meaning on a position $\mathbf{u}$ superimposed with □ such that the position $\mathbf{u} + (0, -1)$ is superimposed with ■, and the position $\mathbf{u} + (-1, 0)$ is superimposed with □, the symbol is $(\#, \rightarrow)$.

8. on a position $\mathbf{u}$ superimposed with $\square$ such that the position $\mathbf{u}+(0,-1)$ is superimposed with $\blacksquare$, and the position $\mathbf{u}+(-1,0)$ is superimposed with $\blacksquare$, the symbol is $(q_0, \#, \rightarrow)$.

As a consequence of these rules, on each computation area is superimposed by a finite part of the space time diagram in which the machine is well initialized. This means that the initial position of the machine is 0, the initial state is q_0, and the initial word written on the tape is $\#^N$. However, it is not possible to make an infinite part of the space-time diagram appear. We will further describe a structure which allows this.

3.4.2 Degenerated behaviors

The structure layer that we presented above is very rigid and does not allow generating organized behaviors. We will need in the following some more flexible structures which in particular exhibit arbitrarily large areas supporting computation. However, when adding other layers with local rules meant to generate some particular behaviors, the flexibility of the structure can imply **degenerated behaviors** in these layers. This means that parts of the language of the subshift contradict dynamical restrictions such as minimality. Our constructions in the next chapters consist in adapting the general scheme of information processing as set in the work of M. Hochman and T. Meyerovitch [HM10], in order to suppress degenerated behaviors, An example of such behavior is the presence, in infinite computation areas, of parts of infinite space-time diagrams that can not appear in any finite space-time diagram.

When adapting this general scheme, the main difficulties are to suppress additional degenerated behaviors triggered by the addition of layers, and by changing the former ones. Most of the degenerated behaviors that we encountered come from **isolated points** of the subshift. An isolated configuration is a configuration such that there exists some finite subset $\mathbb{U}$ of $\mathbb{Z}^d$ such that if any other configuration agrees with this one on $\mathbb{U}$, then the two are equal. In this case, the pattern can not appear in any other configuration.

In this case, the degenerated behavior is a (possibly infinite) part $\mathcal{P}$ of the language of the subshift such that there exists an isolated point x of the subshift, such that for all $p \in \mathcal{P}$, p can appear only in the configurations $\sigma^{\mathbf{u}}(x)$, $\mathbf{u} \in \mathbb{Z}^d$.

Let us present some example of $\mathbb{Z}^2$-SFT constructed as a product of layers where appears some degenerated behavior due to isolated points. We describe how to transform the synchronization layer in order to suppress this behavior.

3.4.2.1 Structure layer

The alphabet of the structure layer is $\mathcal{A}_1 = \left\{\boxed{\nwarrow}, \boxed{\searrow}, \boxed{\nearrow}, \boxed{\swarrow}, \boxed{\rightarrow}, \boxed{\leftarrow}, \boxed{\uparrow}, \boxed{\downarrow}, \blacksquare\right\}$, and the *rules* consist in forbidding the 3-blocks which are not sub-pattern of the pattern on Figure 3.7.

The induced global behavior is that all the configurations in this layer are limit of sequence of configurations as on 3.8, hence they are amongst the following types:

- only one type of symbols covering the plane.
- two half planes, each one colored with a unique symbol, and on the border, the two symbols respect the local rules.

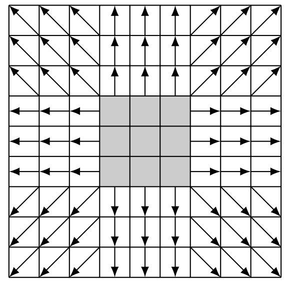


Figure 3.7: The forbidden patterns are the 3-blocks that are not sub-pattern of this one.

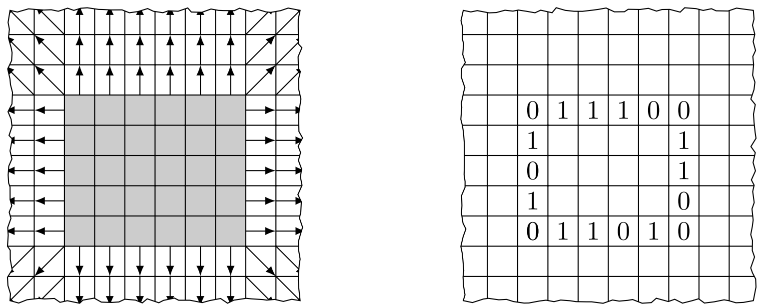


Figure 3.8: Typical configuration in the structure layer, and a possible configuration on this structure in the information layer.

- four quarters of a plane colored with a unique symbol, which respect of the local rules on the border.
- nine areas: eight infinite areas colored with a unique arrow symbol and one fine area colored with gray.
- three areas separated by a thick line or column.

As a consequence, at this point, there is no degenerated behavior due to an isolated configuration. Indeed, any configuration is the limit of a sequence of configurations in the the fourth type.

3.4.2.2 Information layer

The information layer has *symbols* $0, 1$, and a blank symbol.

The *localization rule* is that the non-blank symbols are superimposed on and only on positions $\mathbf{u}$ colored with such that at least one of the positions $\mathbf{u} \pm (0,1)$, $\mathbf{u} \pm (1,0)$ is not colored with .

3.4.2.3 Synchronization layer

Here is a *first attempt* of synchronization layer.

It has *symbols* $0, 1$, and a blank symbol.

The *rules* are

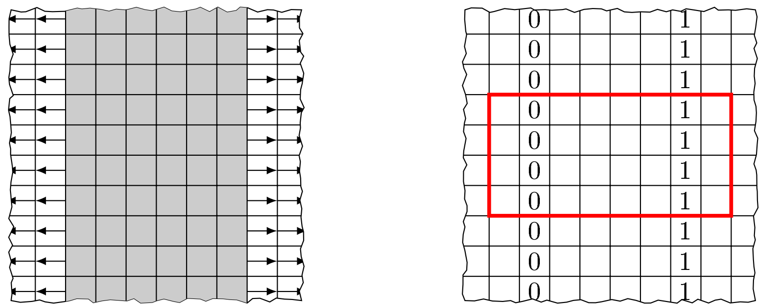


Figure 3.9: Example of degenerated behavior after wire synchronization.

- **Localization:** that the non-blank symbols are superimposed on and only on positions $\mathbf{u}$ colored with such that at least one of the positions $\mathbf{u} \pm (0,1)$, $\mathbf{u} \pm (1,0)$ is not colored with
- **Signal:** two such adjacent positions have the same symbol
- **Synchronization:** the information layer symbol is equal to the synchronization one.

Remark 22. *In this example, we can do the construction without synchronization layer. However, we present the construction with this layer in order to respect the general scheme.*

The *global behavior* is that on a configuration with nine, four or two areas, the information symbols are all equal. On a configuration with three areas, the two separating columns or lines are superimposed with a symbol synchronized along the column or line.

3.4.2.4 Induced degenerated behavior

In the global behavior induced by the synchronization rules the symbols on two different columns or lines of bits in a configuration having three infinite areas can be different, as on Figure 3.9.

This is an example of *degenerated behavior due to isolated points*, since this configuration can not be approximated by a sequence of other configurations. Indeed, no other configuration contains the pattern surrounded on Figure 3.9 than the configuration presented on this figure.

In order to solve this, one can change the localization rule and set that the non-blank symbols are superimposed on and only on positions colored gray in the structure layer.

Hence the effect of the synchronization rules is different. One can call the previous synchronization **wire synchronisation** and this second one **diffusive synchronization**, by analogy with biology [AFSM95]. In our context, the use of diffusive synchronization, as an optimal way to use the space, has an effect on dynamical properties of the system.

3.5 Robinson subshift - a rigid version

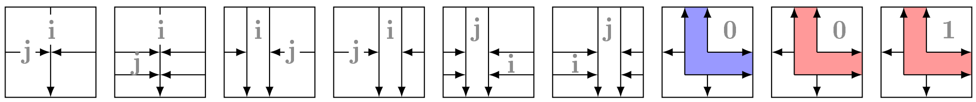
The Robinson subshift was constructed by R. Robinson [Rob71] in order to prove undecidability results. It has been used by in other constructions of subshifts of finite type as a structure layer [PS15].

In this section, we present a version of this subshift which is adapted to constructions under the dynamical constraints that we consider. In order to understand this section, it is preferable to read before the description of the Robinson subshift done in [Rob71]. Some results are well known and we don't give a proof. We refer instead to the initial article of R. Robinson.

Let us denote X_{adR} this subshift, which is constructed as the product of two layers. We present the first layer in Section 3.5.1, and then describe some hierarchical structures appearing in this layer in Section 3.5.2. In Section 3.5.3, we describe the second layer. This layer allows adding rigidity to the first layer, in order to enforce dynamical properties. These properties will be proved in the corresponding chapters.

3.5.1 Robinson layer

The first layer has the following *symbols*, and their transformation by rotations by $\frac{\pi}{2}$, π or $\frac{3\pi}{2}$:

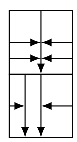


The symbols i and j can have value $0,1$ and are attached respectively to vertical and horizontal arrows. In the text, we refer to this value as the value of the $0,1$-**counter**. In order to simplify the representations, these values will often be omitted on the figures.

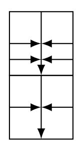
In the text we will often designate as **corners** the two last symbols. The other ones are called **arrows symbols** and are specified by the number of arrows in the symbol. For instance a six arrows symbols are the images by rotation of the fifth and sixth symbols.

The *rules* are the following ones:

1. the outgoing arrows and incoming ones correspond for two adjacent symbols. For instance, the pattern

   is forbidden, but the pattern

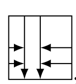
   

   is allowed.

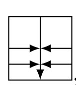
2. in every 2×2 square there is a blue symbol and the presence of a blue symbol in position $\mathbf{u} \in \mathbb{Z}^2$ forces the presence of a blue symbol in the positions $\mathbf{u}+(0,2), \mathbf{u}-(0,2), \mathbf{u}+(2,0)$ and $\mathbf{u}-(2,0)$.

3. on a position having mark (i,j), the first coordinate is transmitted to the horizontally adjacent positions and the second one is transmitted to the vertically adjacent positions.

4. on a six arrows symbol, like

   ,

   or a five arrow symbol, like

   ,

   the marks i and j are different.

The Figure 3.10 shows some pattern in the language of this layer. The subshift on this alphabet and generated by these rules is denoted X_R: this is the Robinson subshift.

The main aspect of this subshift is the following property:

Definition 3.22. *A $\mathbb{Z}^d$-subshift X is said aperiodic when for all configuration x in the subshift, and $\boldsymbol{u} \in \mathbb{Z}^d \backslash (0,0)$,*

$$\sigma^{\boldsymbol{u}}(x) \neq x.$$

Theorem 3.23 ([Rob71])**.** *The subshift X_R is non-empty and aperiodic.*

In the following, we state some properties of this subshift. The proofs of these properties can also be found in [Rob71].

3.5.2 Hierarchical structures

In this section we describe some observable hierarchical structures in the elements of the Robinson subshift.

Let us recall that for all $d \geq 1$ and $k \geq 1$, we denote $\mathbb{U}_k^{(d)}$ the set $[\![0, k-1]\!]^d$.

3.5.2.1 Finite supertiles

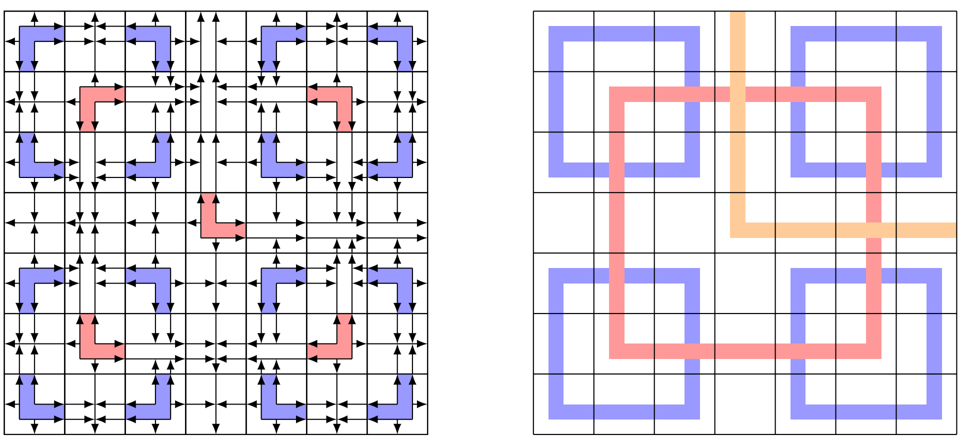
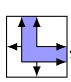
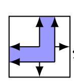
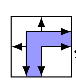
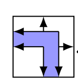
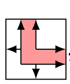
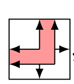
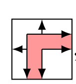
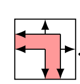


Figure 3.10: The south west order 2 supertile denoted $St_{sw}(2)$ and petals intersecting it.

Let us define by induction the south west (resp. south east, north west, north east) **supertile of order** $n \in \mathbb{N}$. For $n = 0$, one has

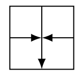
$St_{sw}(0) =$ , $St_{se}(0) =$ , $St_{nw}(0) =$ , $St_{ne}(0) =$ .

For $n \in \mathbb{N}$, the support of the supertile $St_{sw}(n+1)$ (resp. $St_{se}(n+1)$, $St_{nw}(n+1)$, $St_{ne}(n+1)$) is $\mathbb{U}^{(2)}_{2^{n+2}-1}$. On position $\mathbf{u} = (2^{n+1}-1, 2^{n+1}-1)$ write

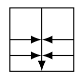
$St_{sw}(n+1)_{\mathbf{u}} =$ , $St_{se}(n+1)_{\mathbf{u}} =$ , $St_{nw}(n+1)_{\mathbf{u}} =$ , $St_{ne}(n+1)_{\mathbf{u}} =$ .

Then complete the supertile such that the restriction to $\mathbb{U}^{(2)}_{2^{n+1}-1}$ (resp. $(2^{n+1},0)+\mathbb{U}^{(2)}_{2^{n+1}-1}$, $(0,2^{n+1})+\mathbb{U}^{(2)}_{2^{n+1}-1}$, $(2^{n+1},2^{n+1})+\mathbb{U}^{(2)}_{2^{n+1}-1}$) is $St_{sw}(n)$ (resp. $St_{se}(n)$, $St_{nw}(n)$, $St_{ne}(n)$).

Then complete the cross with the symbol



or with the symbol



in the south vertical arm with the first symbol when there is one incoming arrow, and the second when there are two. The other arms are completed in a similar way. For instance, Figure 3.10 shows the south west supertile of order two.

Proposition 3.24 ([Rob71])**.** *For all configuration x, if an order n supertile appears in this configuration, then there is an order $n+1$ supertile, having this order n supertile as sub-pattern, which appears in the same configuration.*

3.5.2.2 Infinite supertiles

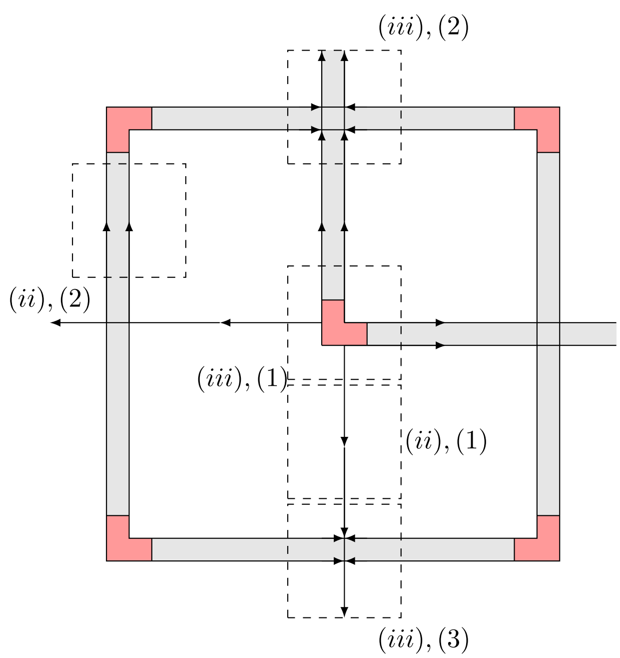


Figure 3.11: Correspondence between infinite supertiles and sub-patterns of order n supertiles. The whole picture represents a schema of some finite order supertile.

Let x be a configuration in the first layer and consider the equivalence relation $\sim_x$ on $\mathbb{Z}^2$ defined by $\mathbf{i} \sim_x \mathbf{j}$ if there is a finite supertile in x which contains $\mathbf{i}$ and $\mathbf{j}$. An **infinite order** supertile is an infinite pattern over an equivalence class of this relation. Each configuration is amongst the following types (with types corresponding with types numbers on Figure 3.11):

(i) A unique infinite order supertile which covers $\mathbb{Z}^2$.

(ii) Two infinite order supertiles separated by a line or a column with only three-arrows symbols (1) or only four arrows symbols (2). In such a configuration, the order n finite supertiles appearing in the two infinite supertiles are not necessary aligned, whereas this is the case in a type (i) or (iii) configuration.

(iii) Four infinite order supertiles, separated by a cross, whose center is superimposed with:

- a red symbol, and arms are filled with arrows symbols induced by the red one. (1)
- a six arrows symbol, and arms are filled with double arrow symbols induced by this one. (2)
- a five arrow symbol, and arms are filled with double arrow symbols and simple arrow symbols induced by this one. (3)

Informally, the types of infinite supertiles correspond to configurations that are limits (for type (ii) infinite supertiles this will be true after alignment [Section 3.5.3]) of a sequence of configurations centered on particular sub-patterns of finite supertiles of order n. This correspondence is illustrated on Figure 3.11. We notice this fact so that it helps to understand how patterns in configurations having multiple infinite supertiles are sub-patterns of finite supertiles.

We say that a pattern p on support $\mathbb{U}$ appears periodically in the horizontal (resp. vertical) direction in a configuration x of a subshift X when there exists some $T > 0$ and $\mathbf{u}_0 \in \mathbb{Z}^2$ such that for all $k \in \mathbb{Z}$,

$$x_{\mathbf{u}_0+\mathbb{U}+kT(1,0)} = p$$

(resp. $x_{\mathbf{u}_0+\mathbb{U}+kT(0,1)} = p$). The number T is called the period of this periodic appearance.

Lemma 3.25 ([Rob71]). *For all n and m integers such that $n \geq m$, any order m supertile appears periodically, horizontally and vertically, in any supertile of order $n \geq m$ with period 2^{m+2}. This is also true inside any infinite supertile.*

3.5.2.3 Petals

For a configuration x of the Robinson subshift some finite subset of $\mathbb{Z}^2$ which has the following properties is called a **petal**.

- this set is minimal with respect to the inclusion,
- it contains some symbol with more than three arrows,
- if a position is in the petal, the next position in the direction, or the opposite one, of the double arrows, is also in it,
- and in the case of a six arrows symbol, the previous property is true only for one pair of arrows.

These sets are represented on the figures as squares joining four corners when these corners have the right orientations.

Petals containing blue symbols are called order 0 petals. Each one intersect a unique greater order petal. The other ones intersect four smaller petals and a greater one: if the intermediate petal is of order $n \geq 1$, then the four smaller are of order $n-1$ and the greatest one is of order $n+1$. Hence they form a hierarchy, and we refer to this in the text as the **petal hierarchy** (or hierarchy).

We usually call the petals valued with 1 **support petals**, and the other ones are called **transmission petals**.

Lemma 3.26 ([Rob71]). *For all n, an order n petal has size $2^{n+1} + 1$.*

We call order n **two dimensional cell** the part of $\mathbb{Z}^2$ which is enclosed in an order $2n + 1$ petal, for $n \geq 0$. We also sometimes refer to the order $2n + 1$ petals as the cells borders.

In particular, order $n \geq 0$ two-dimensional cells have size $4^{n+1} + 1$ and repeat periodically with period 4^{n+2}, vertically and horizontally, in every cell or supertile having greater order.

See an illustration on Figure 3.10.

3.5.3 Alignment positioning

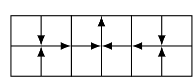
If a configuration of the first layer has two infinite order supertiles, then the two sides of the column or line which separates them are non dependent. The two infinite order supertiles of this configuration can be shifted vertically (resp. horizontally) one from each other, while the configuration obtained stays an element of the subshift. This is an obstacle to dynamical properties such as minimality or transitivity, since a pattern which crosses the separating line can not appear in the other configurations. In this section, we describe additional layers that allow aligning all the supertiles having the same order and eliminate this phenomenon.

Here is a description of the second layer:

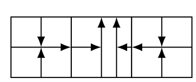
Symbols: nw, ne, sw, se, and a blank symbol.

The *rules* are the following ones:

- **Localization:** the symbols nw, ne ,sw and se are superimposed only on three arrows and five arrows symbols in the Robinson layer.
- **Induction of the orientation:** on a position with a three arrows symbol such that the long arrow originate in a corner is superimposed a symbol corresponding to the orientation of the corner.
- **Transmission rule:** on a three or five arrows symbol position, the symbol in this layer is transmitted to the position in the direction pointed by the long arrow when the Robinson symbol is a three or five arrows symbol with long arrow pointing in the same direction.
- **Synchronization rule:** On the pattern

  or

  in the Robinson layer, if the symbol on the left side is ne (resp. se), then the symbol on the right side is nw (resp. sw). On the images by rotation of these patterns, we impose similar rules.
- **Coherence rule:** the other pairs of symbols are forbidden on these patterns.

Global behavior: the symbols ne, nw, sw, se designate orientations: north east, north west, south west and south east. We will re-use this symbolisation in the following. The localization rule implies that these symbols are superimposed on and only on straight paths connecting the corners of adjacent order n cells for some integer n.

The effect of transmission and synchronization rules is stated by the following lemma:

Lemma 3.27. *In any configuration x of the subshift X_{adR}, any order n supertile appears periodically in the whole configuration, with period 2^{n+2}, horizontally and vertically.*

Proof.

- This property is true in an infinite supertile: this is the statement of Lemma 3.25. Hence the statement is true in a type (i) configuration. This is also true in a type (iii) configuration, since the infinite supertiles are aligned, and that the positions where the order n supertiles appear are the same in any infinite supertile. This statement uses the property that an order n supertile forces the presence of an order $n+1$ one.

- Consider a configuration of the subshift X_{adR} which is of type (ii). Let us assume that the separating line is vertical, the other case being similar. In order to simplify the exposition we assume that this column intersects $(0,0)$.

  1. **Positions of the supertiles along the infinite line:**
     From Lemma 3.25, there exists a sequence of numbers $0 \leq z_n < 2^{n+2}-1$ and $0 \leq z'_n < 2^{n+2}-1$ such that for all $k \in \mathbb{Z}$, the orientation symbol on positions $(-1, z_n + k.2^{n+2})$ (in the column on the left of the separating one) is *se* and the orientation symbols on positions $(1, z'_n + k.2^{n+2})$ is *sw*. The symbol on positions $(-1, z_n + k.2^{n+2} + 2^{n+1})$ is then *ne* and is *nw* on positions on positions $(1, z'_n + k.2^{n+2} + 2^{n+1})$: this comes from the fact that an order n petal has size $2^{n+1}+1$.

     Let us prove that for all n, $z_n = z'_n$. This means that the supertiles of order n on the two sides of the separating line are aligned.

  2. **Periodicity of these positions:**
     Since for all n, there is a space of 2^n columns between the rightmost or leftmost order n supertile in a greater order supertile and the border of this supertile (by a recurrence argument), this means that the space between the rightmost order n supertiles of the left infinite supertile and the leftmost order n supertile of the right infinite supertile is $2^{n+1}+1$. Since two adjacent of these supertiles have opposite orientations, this implies that each supertile appears periodically in the horizontal direction (and hence both horizontal and vertical directions) with period 2^{n+2}.

  3. **The orientation symbols force alignment:**
     Assume that there exists some n such that $z_n \neq z'_n$. Since $\{z_n + k.2^{n+2},\ (n,k) \in \mathbb{N} \times \mathbb{Z}\} = \mathbb{Z}$, this implies that there exist some $m \neq n$ and some k, k' such that

     $$z_n + k.2^{n+2} = z'_m + k'.2^{m+2}.$$

     One can assume without loss of generality that $m < n$, exchanging m and n if necessary. Then the position $(-1, z_n + k.2^{n+2} + 2^{n+1})$ has orientation symbol equal to *ne*. As a consequence, the position $(1, z'_m + k'.2^{m+2} + 2^{n-m-1}.2^{m+2})$ has the same symbol. However, by definition, this position has symbol *se*: there is a contradiction. This situation is illustrated on Figure 3.12.

□

3.5.4 Completing blocks

Let $\chi : \mathbb{N}^* \to \mathbb{N}^*$ such that for all $n \geq 1$,

$$\chi(n) = \left\lceil \log_2(n) \right\rceil + 4.$$

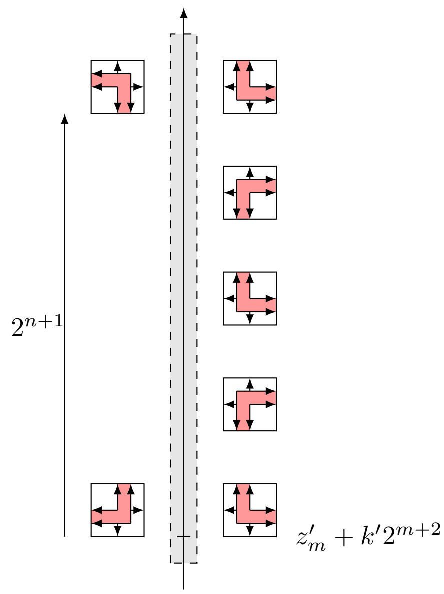


Figure 3.12: Schema of the proof. The separating line is colored gray.

Let us also denote χ' the function such that for all $n \geq 1$,

$$\chi'(n) = \left\lceil \frac{\lceil \log_2(n) \rceil}{2} \right\rceil + 2.$$

The following lemma will be extensively used in the following of this text, in order to prove dynamical properties of the constructed subshifts:

Lemma 3.28. *For all $n \geq 1$, any n-block in the language of X_{adR} is a sub-pattern of some order $\chi(n)$ supertile, and is a sub-pattern of some order $\chi'(n)$ order cell.*

Proof.

1. **Completing into an order $2^{\lceil \log_2(n) \rceil + 1} - 1$ block:**

   Consider some n-block p that appears in some configuration x of the SFT X_{adR}. We can complete it into a $2^{\lceil \log_2(n) \rceil + 1} - 1$ block, since $2^{\lceil \log_2(n) \rceil + 1} - 1 \geq 2n - 1 \geq n$ for all $n \geq 1$.

2. **Intersection with four order $\lceil \log_2(n) \rceil$ supertiles:**

   From the periodic appearance property of the order $\lceil \log_2(n) \rceil$ supertiles in each configuration, this last block intersects at most four supertiles having this order. Let us complete p into the block whose support is the union of the supports of the supertiles and the cross separating these.

3. **Possible patterns after this completion according to the center symbol:**

   Since this pattern is determined by the symbol at the center of the cross and the orientations of the supertiles, the possibilities for this pattern are listed on Figure 3.13, Figure 3.14 and Figure 3.15. Indeed, when the orientations of the supertiles are like on Figure 3.13, each of the supertiles forcing the presence of an order $\lfloor \log_2(n) \rfloor + 1$ supertile, the center is a red corner. When the orientations of the supertiles are like on Figure 3.14, the center of the block can not be superimposed with a red corner since the two west supertiles force an order $\lfloor \log_2(n) \rfloor + 1$

supertile, as well as the two east supertiles. This forces a non-corner symbol on the position considered.

For type $4, 5, 9, 10$ patterns, there are two possibilities: the values of the two arms of the central cross are equal or not. Hence the notation $4, 4'$, where $4'$ designates the case where the two values are different.

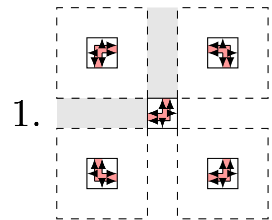


Figure 3.13: Possible orientations of four neighbor supertiles having the same order (1).

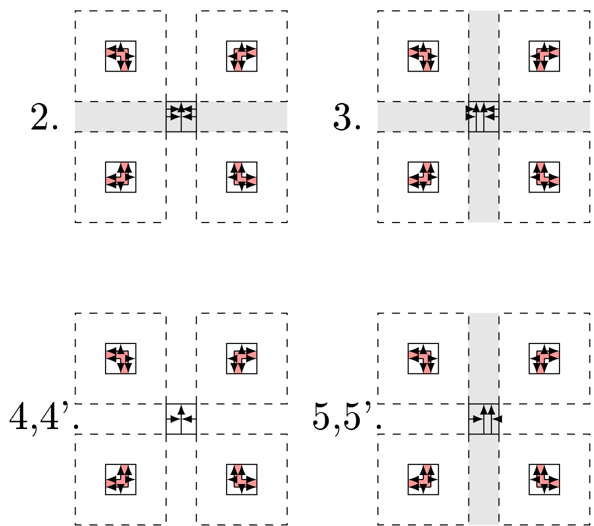


Figure 3.14: Possible orientations of four neighbor supertiles having the same order (2).

One completes the alignment layer on p according to the restriction of the configuration x.

On these patterns, the value of symbols on the cross is opposed to the value of the symbols on the crosses of the four supertiles composing it.

4. **Localization of these patterns as part of a greater cell:**

   The way to complete the obtained pattern is described as follows:

   (a) When the pattern is the one on Figure 3.13, this is an order $\lfloor \log_2(n) \rfloor + 1$ supertile and the statement is proved. Indeed, any order $\lceil \log_2(n) \rceil + 1$ supertile is a sub-pattern of any order $\lfloor \log_2(n) \rfloor + 4$ one.

   (b) One can see the patterns on Figure 3.14 and Figure 3.15 in an order $\lfloor \log_2(n) \rfloor + 1$, $\lfloor \log_2(n) \rfloor + 2$, $\lfloor \log_2(n) \rfloor + 3$, or $\lfloor \log_2(n) \rfloor + 4$ supertile, depending on how was completed the initial

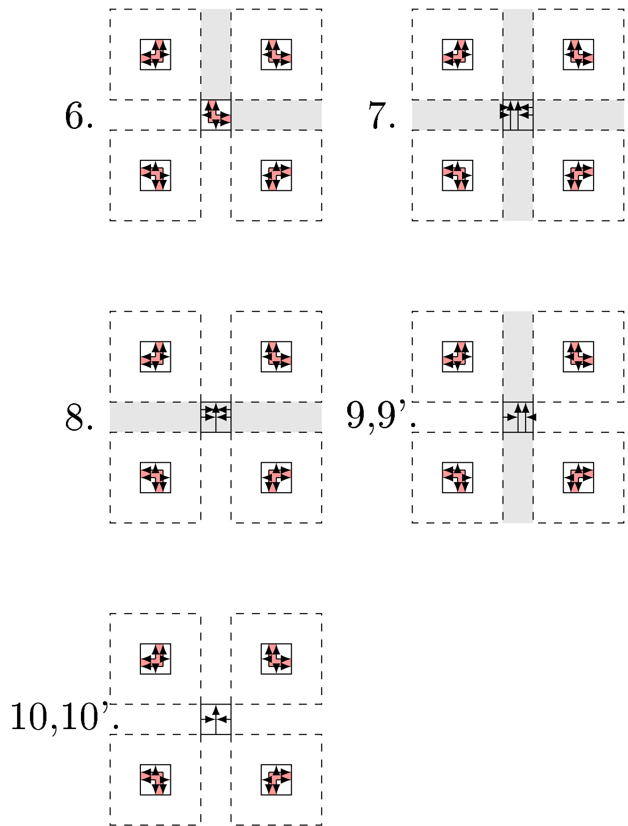


Figure 3.15: Possible orientations of four neighbor supertiles having the same order (3).

pattern thus far (this correspondance is shown on Figure 3.16), hence a sub-pattern of an order $\lfloor \log_2(n) \rfloor + 4$ supertile.

The orientation of the greater order supertiles implied in this completion are chosen according to the symbols of the alignment layer. This layer is then completed.

5. This implies that any n-block is the sub-pattern of an order $2(\left\lceil \frac{1}{2} \lceil \log_2(n) \rceil \right\rceil + 2) + 1$ supertile, which is included into an order $\left\lceil \frac{1}{2} \lceil \log_2(n) \rceil \right\rceil + 2$ cell.

$\square$

Proposition 3.29. *The subshift X_{adR} is a minimal SFT.*

Remark 23. *Let use note that the existence of a minimal $\mathbb{Z}^2$-SFT is already known, and was proved the first time by S. Mozes.*

Proof. Any cell appears in all the configurations of this subshift. Since any pattern is sub-pattern of a cell, it then appears in all the configurations. Hence X_{adR} is a minimal SFT. $\square$

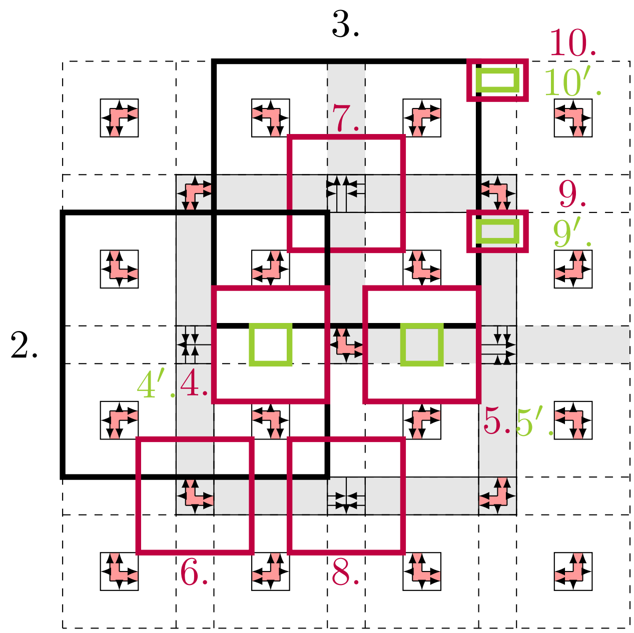


Figure 3.16: Illustration of the correspondance between patterns of Figure 3.14 and parts of a supertile.

Chapter 4

Periodic points and entropy of bidimensional SFT under block gluing restriction with gap function

Sommaire

This chapter is a partial reproduction of the article *Block gluing intensity of bidimensional SFT: entropy and periodic points* [GS17a].

Abstract

In this chapter, we introduce the notion of block gluing with gap function. We study the interplay between these properties and properties related to computability for $\mathbb{Z}^2$-SFTs. In particular, we prove that there exist linearly block gluing SFTs which are aperiodic and that all the non-negative Π_1-computable numbers can be realized as entropy of linearly block gluing $\mathbb{Z}^2$-subshifts of finite type. As linearly block gluing implies transitivity, this last point answers a question asked in [HM10] about the characterization of the entropies of transitive subshift of finite type.

4.1 Introduction

It appeared recently that it is possible to understand many dynamical properties of multidimensional subshifts of finite type through characterizations results, beginning with the entropy [HM10]. This interplay between dynamical properties and computability or decidability properties was shown by a lot of recent works: a characterization of the projective sub-action of a SFT [Hoc09, AS13, DRS12], measure of the computationnally simplest configurations with Medvedev degrees [Sim14] and sets of Turing degrees [JV13], characterization of the possible sets of periods in terms of complexity theory [JV15], etc.

The importance of computability considerations in these models has been clearly established for decades now. A new direction is to see if dynamical properties can prevent embedding universal computing.

One can observe for instance this phenomenon for some mixing-like properties. To be more precise, the result proved by M. Hochman and T. Meyerovitch [HM10] states that the set of numbers that are entropy of a multidimensional SFT is exactly the set of non-negative Π_1-computable real numbers. When the SFT is strongly irreducible, the entropy becomes computable. Only some step results towards a characterization are known [PS15]. We interpret this obstruction as a reduction of the computational power of the model under this restriction, which is manifested by the reduction of possible entropies.

In [PS15] the authors studied SFT which are block gluing. This means that there exists a constant c such that for any pair of square blocks in the language of the subshift, the pattern obtained by gluing these two blocks in any way respecting a distance c between them is also in the language.

We propose in this chapter to study similar properties that consist in imposing the gluing property when the two blocks are spaced by $f(n)$, where n is the length of two square blocks having the same size, and $f : \mathbb{N} \to \mathbb{N}$.

We got interested in the influence of this property on the possibility of non-existence of periodic orbits (which is the first ingredient for embedding computations in SFT), and on the set of possible entropies, according to the gap function f. We observed two regimes:

- **Sub-logarithmic:** if $f \in o(\log(n))$ and $f \leq id$, then the set of periodic orbits of any f-block gluing SFT is dense (Proposition 4.13) and its language is decidable. Moreover, the entropy is computable (Proposition 4.27).

- **Linear:**

  - There exists $f \in O(n)$ such that there exists an aperiodic f-block gluing SFT (Theorem 1.1).
  - There exists $f \in O(n)$ such that there exists a f-block gluing SFT with non decidable language (Proposition 4.15).
  - The possible entropies of f-block gluing SFT with $f \in O(n)$ are the right recursively enumerable non-negative real numbers (Theorem 1.2).

The aim was to understand the limit between these two regimes. We extended to the sub-logarithmic regime some results on block gluing subshifts of [PS15]. We adapted the realization result of [HM10] to block gluing with linear gap function. This adaptation involved more complex and structured constructions. The mechanisms introduced exhibit an attractive analogy with very simple molecular biology objects. This analogy is present in the words that we use in to describe the construction, that are useful to visualize and understand the construction. In particular, the applied constraint has resulted in the centralization and fixation of information (called DNA) in the centers (called nuclei) of the computing units (called cells). The computing machines communicate using error signals to have access to this information, which codes for its behavior.

This leads to the intuition that this type of results could be read in order to understand the principles of *information processing systems*, as *"mixing implies centralization of the information"*.

The chapter is organized as follows:

- In Section 4.2 we define the notion of block gluing with gap function, and provide many examples.

- Section 4.3 is devoted to study of the existence and density of periodic points under block gluing constraint.

- Section 4.4 is devoted to the entropy.

4.2 Notion of block gluing with gap function

In this section, we introduce the notion of block gluing with intensity function. Then we give some examples of subshifts of finite type which are block gluing for some particular intensity functions (f is constant, $f = \log$ and f is linear).

In this chapter, a configuration $x \in \mathcal{A}^{\mathbb{Z}^2}$ is said to be (doubly) **periodic** if there exist some $m, n > 0$ such that

$$\sigma^{(m,0)}(x) = \sigma^{(0,n)}(x) = x.$$

In all the following, we say that a subset Λ of $\mathbb{Z}^d$ is **connected** when for any two of its elements $\mathbf{l}^1, \mathbf{l}^2$, there exists $n \geq 1$ and a function

$$\varphi : [\![0, n]\!] \to \Lambda$$

such that $\varphi(0) = \mathbf{l}^1$, $\varphi(1) = \mathbf{l}^2$, and for all $t \in [\![0, n-1]\!]$,

$$\varphi(t+1) - \varphi(t) \in \{\mathbf{e}^1, ..., \mathbf{e}^d\}.$$

4.2.1 Block gluing notions

4.2.1.1 Definitions

In this section, X is a subshift on the alphabet $\mathcal{A}$ and $f : \mathbb{N} \to \mathbb{N}$ is a non decreasing function. We denote $||.||_\infty$ the norm defined by

$$||\mathbf{i}||_\infty = \max\{\mathbf{i}_1, \mathbf{i}_2\}$$

for all $\mathbf{i} \in \mathbb{Z}^2$. We denote d_∞ the associated distance function.

Definition 4.1. *Let $n \in \mathbb{N}$ be an integer. The **gluing set** in the subshift X of some n-block p relative to some other n-block q is the set of $\boldsymbol{u} \in \mathbb{Z}^2$ such that there exists a configuration in X where q appears in position $(0,0)$ and p appears in position $\boldsymbol{u}$ (see Figure 4.1). This set is denoted $\Delta_X(p,q)$. Formally*

$$\Delta_X(p,q) = \left\{\boldsymbol{u} \in \mathbb{Z}^2 : \ \exists x \in X \text{ such that } x_{[\![0,n-1]\!]^2} = q \text{ and } x_{\boldsymbol{u}+[\![0,n-1]\!]^2} = p\right\}$$

*When the intersection of the sets $\Delta_X(p,q)$ for (p,q) ordered pairs of n-blocks is non empty, we denote this intersection $\Delta_X(n)$. This set is called the **gluing set of n-blocks** in X.*

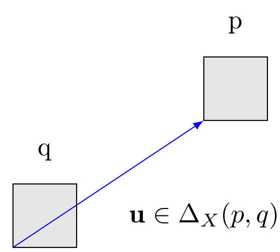


Figure 4.1: Illustration of Definition 4.1.

Definition 4.2. *A subshift X is said to be f**-block transitive** if for all $n \in \mathbb{N}$ one has*

$$\Delta_X(n) \cap \left\{\boldsymbol{u} \in \mathbb{Z}^2 : ||\boldsymbol{u}||_\infty \leq n + f(n)\right\} \neq \emptyset.$$

*The function f is called the **gap function**.*

Remark 24. *The condition on the vectors $\boldsymbol{u} \in \mathbb{Z}^2$ is $||\boldsymbol{u}||_\infty \leq n + f(n)$ since we consider the distance between the positions where the patterns appear instead of the space between them.*

Definition 4.3. *A subshift X is said to be f **net gluing** if there exists a function $\boldsymbol{u} : \mathcal{L}(X)^2 \to \mathbb{Z}^2$ and a function $\widetilde{f} : \mathcal{L}(X)^2 \to \mathbb{N}$ such that for all $n \in \mathbb{N}$ and for all n-blocks p and q,*

$$\boldsymbol{u}(p,q) + (n + \widetilde{f}(p,q))(\mathbb{Z}^2 \backslash \{0\}) \subset \Delta_X(p,q)$$

and

$$\max_{p,q \in \mathcal{L}_n(X)} \widetilde{f}(p,q) \leq f(n).$$

Remark 25. *This property is different from quasiperiodicity properties in the sense that the configuration where two patterns appears can depend on the relative position of the two patterns.*

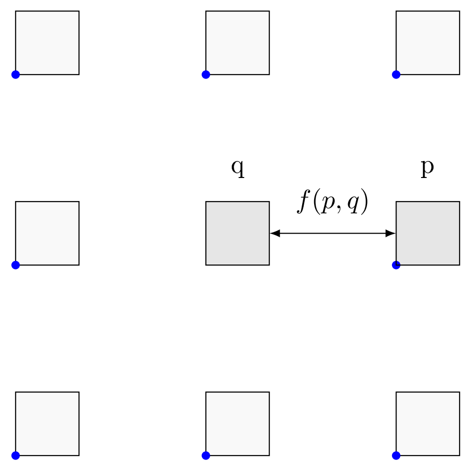


Figure 4.2: Illustration of Definition 4.3. The blue points designate elements of the gluing set of p relatively to q in X.

Definition 4.4. *A subshift X is f-**block gluing** when*

$$\left\{\boldsymbol{u} \in \mathbb{Z}^2, ||\boldsymbol{u}||_\infty \geq f(n) + n\right\} \subset \Delta_X(n).$$

For any function f, one has

$$f\text{-block gluing} \Longrightarrow f\text{-net gluing} \Longrightarrow f\text{-block transitive}$$

A subshift is said $O(f)$**-block gluing** (resp. $O(f)$**-net gluing**, $O(f)$**-block transitive**) if it is g (block gluing) (resp. g-net gluing, g-block transitive) for a function $g : \mathbb{N} \to \mathbb{N}$ such that there exists $C > 0$ such that $g(n) \leq C\, f(n)$ for all $n \in \mathbb{N}$. A property verified on the class of $O(f)$ block gluing (resp. g net gluing, g block transitive) subshifts is said to be **sharp** if the property is false for all $h \in o(f)$ (this means that for all $\epsilon > 0$ there exists n_0 such that $h(n) \leq \epsilon f(n)$ for all $n \geq n_0$).

A subshift is **linearly block gluing** (resp. **linearly net gluing**, **linearly transitive**) if it is $O(n)$ block gluing (resp. $O(n)$ net gluing, $O(n)$ block transitive).

4.2.1.2 Equivalent definition

The following proposition gives an equivalent definition for linear block gluing and net gluing subshifts using some exceptional values:

Proposition 4.5. *A subshift $X \subset \mathcal{A}^{\mathbb{Z}^2}$ is linearly block gluing if and only if there exists a function $f \in O(n)$, $c \geq 2$ an integer and $m \in \mathbb{N}$ such that*

$$\{\boldsymbol{u} \in \mathbb{Z}^2, ||\boldsymbol{u}||_\infty \geq f(c^l + m) + c^l + m\} \subset \Delta_X(c^l + m) \qquad \forall l \geq 0.$$

A similar assertion is true for net gluing.

Proof. Clearly a linear-block gluing subshift verifies this property. Reciprocally, let p and q be two n-blocks, and consider $l(n) = \lceil \log_c(n-m) \rceil$, where for all real number x, $\lceil x \rceil$ designates the smallest integer greater than x. Consider p' and q' some $c^{l(n)} + m$-blocks whose restrictions on $[\![0, n-1]\!]^k$ are respectively p and q. The set $\Delta_X(p', q')$ contains $\{\mathbf{u} \in \mathbb{Z}^2, ||\mathbf{u}||_\infty \geq f(c^{l(n)} + m) + c^{l(n)} + m\}$. As a consequence, $\Delta_X(p,q)$ contains $\{\mathbf{u} \in \mathbb{Z}^2, ||\mathbf{u}||_\infty \geq g(n) + n\}$, where $g(n) = f(c^{l(n)} + m) + c^{l(n)} - n + m$. Since $c^{l(n)} \leq c * (n + |m|)$, the function g is in $O(n)$, hence X is $O(n)$-block gluing. $\square$

The proofs of block gluing and net gluing properties often rely on this proposition. In our construction, we first complete patterns into patterns over cells, whose sizes are given by an exponential sequence as in the statement of the proposition. Then we prove the gluing property for two cells.

4.2.1.3 Block gluing and morphisms

The following proposition shows that a factor of a block gluing (resp. net gluing) subshift is also block gluing (resp. net gluing) and gives a precise gap function.

Proposition 4.6. *Let $\varphi : X \to Y$ be some onto r-block map between two $\mathbb{Z}^2$-subshifts, and $f : \mathbb{N} \to \mathbb{N}$ a non decreasing function. If the subshift X is f-block gluing (resp. f- net gluing), then Y is g-block gluing (resp. g-net gluing) where $g : n \longmapsto f(n+2r)+2r$.*

Proof. Denote $\overline{\varphi} : \mathcal{A}_X^{[\![-r,r]\!]^2} \to \mathcal{A}_Y$ the local rule of φ.

1. **Relation between gluing sets:**

   Let p', q' be two n-blocks in the language of Y. There exist p and q two $(n+2r)$-blocks in the language of X such that p' and q' are respectively the image of p and q by $\overline{\varphi}$. Let $\mathbf{u} \in \Delta_X(p,q)$. There exists $x \in X$ such that $x_{[\![0,n+2r-1]\!]^2} = p$, and $x_{\mathbf{u}+[\![0,n+2r-1]\!]^2} = q$. Applying φ to $\sigma^{r(1,1)}(x)$, we obtain some $y \in Y$ such that $y_{[\![0,n-1]\!]^k} = p'$, and $y_{\mathbf{u}+[\![0,n-1]\!]^k} = q'$. We deduce that

   $$\Delta_X(p,q) \subset \Delta_Y(p',q') \quad \text{so} \quad \Delta_X(n+2r) \subset \Delta_Y(n).$$

2. **Consequence of the gluing property:**

   Thus if X is f-block gluing then Y is g-block gluing where $g : n \longmapsto f(n+2r)+2r$.

   If X is f-net gluing, then the gluing set of two $(2n+r)$-blocks p, q contains

   $$\mathbf{u}(p,q) + (n+2r+\widetilde{f}(p,q))(\mathbb{Z}^2 \backslash \{(0,0)\}),$$

   such that $\widetilde{f}(p,q) \leq f(n+2r)$. Hence the gluing set of p', image of p by $\overline{\varphi}$, relative to q', image of q by $\overline{\varphi}$, in Z contains this set. One deduces that Y is g-net gluing where $g : n \longmapsto f(n+2r)+2r$.

□

We deduce that the classes of subshifts defined by these properties are invariant of conjugacy under some assumption on f. The set functions that verify this assumption includes all the possible gap functions we already know:

Corollary 4.7. *Let f be some non decreasing function. If for all $r \in \mathbb{N}$, there is a constant C such that for all $n \geq 0$, $Cf(n) \geq f(n+2r)$ then the following classes of subshifts are invariant under conjugacy: $O(f)$-block transitive, $O(f)$-net gluing, $O(f)$-block gluing, sharp $O(f)$-net gluing and sharp $O(f)$-block gluing subshifts.*

In particular it is verified when f is constant or $n \mapsto n^k$ with $k > 0$ or $n \mapsto e^n$ or $n \mapsto \log(n)$.

4.2.2 Some examples

We say that two blocks p, q having respective supports $\mathbb{U}, \mathbb{V}$ are **spaced by distance** k when

$$\min_{\mathbf{u} \in \mathbb{U}} \min_{\mathbf{v} \in \mathbb{V}} ||\mathbf{u} - \mathbf{v}||_\infty \geq k.$$

Since the subshifts that we consider in this chapter are bi-dimensional, this means that there are at least k column or at least k lines between the two blocks.

4.2.2.1 First examples

We present here some examples of block gluing SFT.

Example 4.8. *Consider the SFT $X_{\texttt{Chess}}$ defined by the following set of forbidden patterns:*

This subshift has two configurations (see Figure 4.3 for an example) and both of them are periodic. It is 1*-net gluing, but not block gluing. Indeed, the gluing set of the pattern* $\blacksquare$ *relatively to itself is*

$$\Delta_{X_{\texttt{Chess}}}(\blacksquare, \blacksquare) = 2\mathbb{Z}^2 \setminus \{(0,0)\} \cup \left(2\mathbb{Z}^2 + (1,1)\right)$$

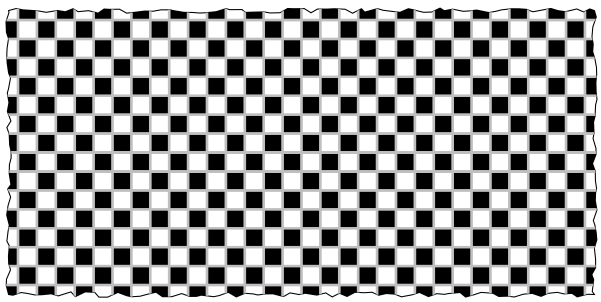


Figure 4.3: An example of configuration of $X_{\texttt{Chess}}$.

Example 4.9. *Consider the SFT $X_{\texttt{Even}}$ defined by the following set of forbidden patterns:*

An example of configuration in this subshift is given in Figure 4.4. This subshift is 1*-block gluing since two blocks in its language can be glued with distance 1, filling the configuration with* $\square$ *symbols.*

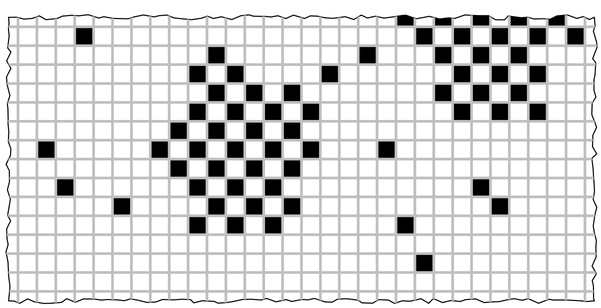
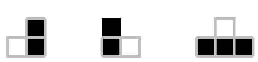


Figure 4.4: An example of configuration of $X_{\texttt{Even}}$.

Example 4.10. *Consider the SFT $X_{\texttt{Linear}}$ defined by the following set of forbidden patterns:*



The local rules imply that if a configuration contains the pattern $\square\blacksquare^n\square$ then it contains $*\square\blacksquare^{n-2}\square*$ just above, where $* \in \{\square, \blacksquare\}$. Thus a configuration of $X_{\texttt{Linear}}$ can be seen as a layout of triangles of made of symbols $\blacksquare$ on a background of $\square$ symbols (an example of configuration is given on Figure 4.5).

This subshift is sharp linearly block gluing. Indeed, consider two n-blocks in its language separated horizontally or vertically by $2n$ cells. They contain pieces of triangles that we complete with the smallest triangle possible, the other symbols of the configuration being all $\square$ symbols. The worst case for gluing two n-blocks is when the blocks are filled with the symbol $\blacksquare$. In this case we can complete each of the two blocks by a triangle which base is constituted by $\blacksquare^{3n}$. Hence every pair of blocks can be glued horizontally and vertically with linear distance. To prove that $X_{\texttt{Linear}}$ is not f-block gluing with $f(n) \in o(n)$, we consider the rectangle

$$\square\blacksquare^n\square$$

that we would like to glue above itself. To do that we need to separate the two copies of this pattern by about $\lceil \frac{n}{2} \rceil$ cells.

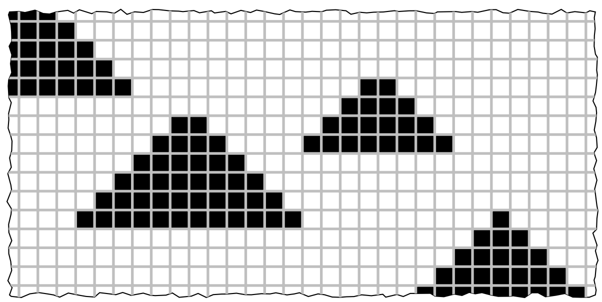


Figure 4.5: An example of configuration of $X_{\texttt{Linear}}$.

4.2.2.2 Linearly net gluing subshifts given by substitutions

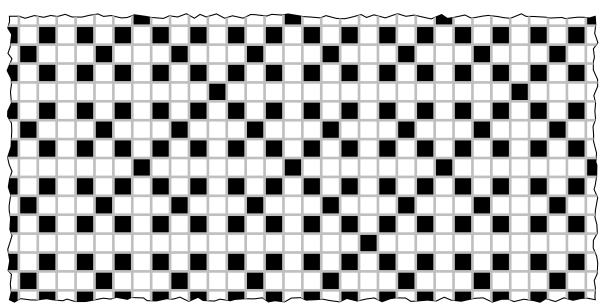


Figure 4.6: A part of a configuration of X_s.

Let $\mathcal{A}$ be a finite alphabet. A **substitution** rule is a map $s : \mathcal{A} \to \mathcal{A}^{\mathbb{U}_m}$, for some $m \geq 1$. This function can be extended naturally on blocks in view to iterate it. The subshift X_s associated to this substitution is the set of configurations such that any pattern appearing in it appears as a sub-pattern of some $s^n(a)$ with $n \geq 0$ and $a \in \mathcal{A}$.

Consider the following substitution s defined by

where an exemple of configuration is given in Figure 4.6. Since ■ appears on position $(0,0)$ in $s(\square)$ and $s(\blacksquare)$, we deduce that for any configuration x, there exists $\mathbf{i}_1 \in [\![0,1]\!]^2$ such that $x_{\mathbf{i}_1+2\mathbb{Z}^2} = \blacksquare$. By induction, for all $n \geq 1$, there exists $\mathbf{i}_n \in [\![0,2^n-1]\!]^2$ such that $x_{\mathbf{i}_n+2^n\mathbb{Z}^2} = s^n(\blacksquare)$. Since every pattern of X_s appears in $s^n(\blacksquare)$ for some $n \in \mathbb{N}$, we deduce that X_s has the linear net-gluing property, using Proposition 4.5.

This argument can be easily generalized for substitution s for which there exists $i \in \mathbb{N}$, a subset $\mathcal{Z} \subset [\![0,m^i-1]\!]^2$ and an invertible map $\nu : \mathcal{A} \to \mathcal{Z}$ such that $a \in \mathcal{A}$ appears on the same position $\nu(a)$ in any pattern of the patterns $s^i(d)$ with $d \in \mathcal{A}$.

4.2.2.3 Intermediate intensities

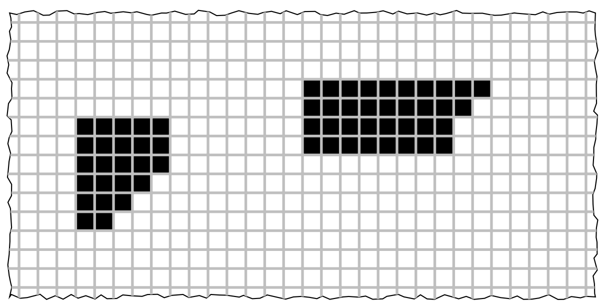


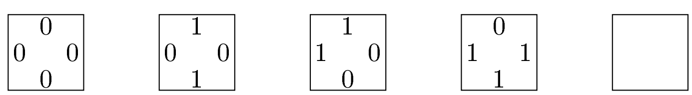
Figure 4.7: An example of configuration that respects the rules of the first layer of $X_{\mathtt{Log}}$.

Here we present an example of block gluing SFT whose gap function is strictly between linear and constant classes.

Consider the SFT $X_{\mathtt{Log}}$ having two layers, with the following characteristics:

Symbols:

The first layer has symbols ■ and □, and the second one the symbols:



The first four of these symbols are thought as coding for the adding machine. Each one contains four symbols: the west one is the initial state of the machine, the east one the forward state, the south one the input letter, and the last symbol is the output.

Local rules:

- **First layer:**

  The following patterns are forbidden in the first layer:

These local rules imply that if a configuration contains the pattern $\square\blacksquare^n\square$ in the first layer, then it contains $\square\blacksquare^n\square$, $\square\blacksquare^{n+1}$, or $\square^{n+2}$ just above. Thus a configuration of the first layer of $X_{\mathtt{Log}}$

can be seen as triangular shapes of symbols $\blacksquare$ on a background of $\square$ symbols (an example of configuration is given in Figure 4.7).

- **Second layer:**
  - the adding machine symbols are superimposed on black squares, the other ones on blank squares.
  - for two adjacent machine symbols, the symbols on the sides have to match.
  - on a pattern $\blacksquare\square$, on the machine symbol over the black square, the east symbol have to be 0.
  - on a pattern $\begin{smallmatrix}\blacksquare & \blacksquare \\ \blacksquare & \square\end{smallmatrix}$, the machine have a south symbol being 0 on the north west black square.

This subshift is sharp $O(log)$-block gluing. Indeed, any two n-blocks in its language can be glued vertically with distance 1. For the horizontal gluing, the worst case for gluing two n-blocks is when the two blocks are filled with black squares and the adding machine symbols on the leftmost column of the blocks are only 1 (thus maximizing the number of lines where the rectangular shape into which we complete the block have to be greater in length than the one just below). In this case, we can complete the block such that each line (from the bottom to the top) is extended from the one below with one $\blacksquare$ symbol on the right when the machine symbol have a 1 on its west side, and adding blank squares to obtain a rectangle. The number of columns added is smaller than the maximal number of bits added by the adding machine to a length n string of $0, 1$ symbols in n steps, which is $O(log(n))$. This means that two n-blocks can be glued horizontally with distance $O(log(n))$. To see that this property is sharp, consider the horizontal gluing of two $1 \times n$ rectangles of black squares, similarly as in the linear case.

Remark 26. *The set of possible tight gap functions of block gluing SFT seems restricted. We don't know for instance if, when f is the square root function, there exist subshifts for which the f block gluing property is tight.*

4.2.3 The rigid version of the Robinson subshift is linearly net gluing

Let us denote $id : \mathbb{N} \to \mathbb{N}$ the function defined by $id(n) = n$ for all n.

Proposition 4.11. *The subshift X_{adR} is $32id$-net gluing, hence linearly net gluing.*

Proof. Let p, q be two n-blocks in the language of X_{adR} with $n \geq 1$. There exists $m \in \mathbb{N}$ such that $2^{m+1} - 1 < n \leq 2^{m+2} - 1$. Hence, there is a supertile S of order $m + 5$ (this comes from the completion result proved in Chapter 3) where p appears. Consider a configuration $x \in X_{adR}$ in which the pattern q appears in position $(0, 0)$. The supertile S appears periodically in x with period $2^{m+6} = 32.2^{m+1} \leq 32n$. Thus the gluing set of p relatively to q in X_{adR} contains a set $\mathbf{u} + 2^{m+6}(\mathbb{Z}^2 \backslash \{(0, 0)\})$ for some $\mathbf{u} \in \mathbb{Z}^2$. Thus X_{adR} is 32id net gluing. □

4.3 Existence of periodic points for f-block gluing SFT

In this section we study the existence of a periodic point in f-block gluing SFT according to the gap function f.

4.3.1 Under some threshold for the gap function, there exist periodic points

In [PS15], the authors show that any constant block gluing SFT admits a periodic point. Using a similar argument, we obtain an upper bound on the gap functions forcing the existence of periodic points.

Proposition 4.12. *Let $X \subset \mathcal{A}^{\mathbb{Z}^2}$ be some SFT having rank $r \geq 2$ which is f-block gluing for some function f. If this function verifies that there exists $n \in \mathbb{N}$ such that*

$$f(n) < \frac{\log_{|\mathcal{A}|}(n-r+2)}{r-1} - r + 2,$$

then X admits a periodic point.

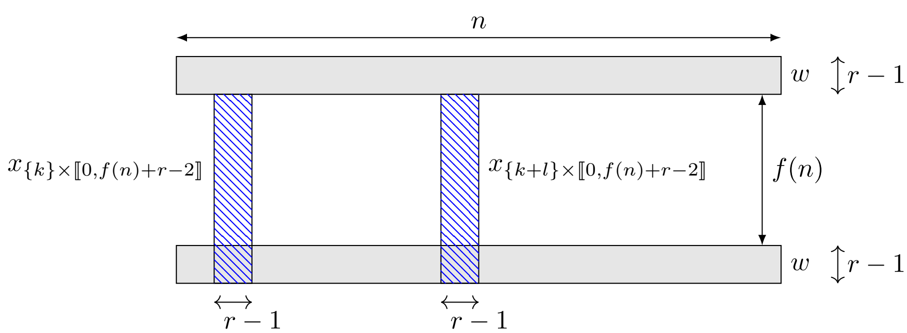


Figure 4.8: If $n-r+2 > |\mathcal{A}|^{(r-1)(f(n)+r-2)}$, it is possible to find a pattern of x for which the horizontal and vertical borders are similar.

Proof. Let w be a $n \times (r-1)$ pattern in the language of X.

1. **Gluing the pattern w over itself:**

   By the f-block gluing property, there exists $x \in X$ such that

   $$x_{[\![0,n-1]\!]\times[\![0,r-2]\!]} = w = x_{[\![0,n-1]\!]\times[\![f(n)+r-1,f(n)+2r-3]\!]}.$$

2. **Taking w long enough, two of the columns in the obtained pattern are equal:**

   Consider the sub-patterns of $x_{[\![0,n-1]\!]\times[\![0,f(n)+r-2]\!]}$ over supports $[\![k,k+r-2]\!]\times[\![0,(f(n)+r-2)]\!]$. There are $n-(r-2)$ of them and the number of possibilities is $|\mathcal{A}|^{(r-1)(f(n)+r-2)}$. Since we have

   $$f(n) < \frac{\log_{|\mathcal{A}|}(n-r+2)}{r-1} - r + 2,$$

   by the pigeon hole principle, there exists $k \in [\![0, n-r+1]\!]$ and $l \geq 1$ such that

   $$x_{[\![k,k+r-2]\!]\times[\![0,f(n)+r-2]\!]} = x_{[\![k+l,k+l+r-2]\!]\times[\![0,f(n)+r-2]\!]}$$

   (see Figure 4.8).

3. **Construction of a periodic configuration:**

   Consider the configuration defined by

$$z_{(il,j(f(n)+r-2))+[\![0,l-1]\!]\times[\![0,f(n)+r-2]\!]} = x_{[\![k,k+l-1]\!]\times[\![0,f(n)+r-2]\!]}$$

   for all $(i,j) \in \mathbb{Z}^2$. It consists in covering $\mathbb{Z}^2$ with the pattern $x_{[\![k,k+l-1]\!]\times[\![0,f(n)+r-2]\!]}$. This configuration is periodic by definition. Moreover, it satisfies the local rules of X. We deduce that $z \in X$. Thus X admits a periodic point.

□

4.3.2 Under some smaller threshold, the set of periodic points is dense and the language is decidable

Using a similar argument as in [PS15], we obtain also an upper bound on the gap functions that force the density of periodic points, and as a consequence the decidability of the language.

Proposition 4.13. *Let X be some f-block gluing $\mathbb{Z}^2$-SFT, where f is a function such that*

$$f(n) \in o(\log(n))$$

and $f \leq id$. Then X has a dense set of periodic points.

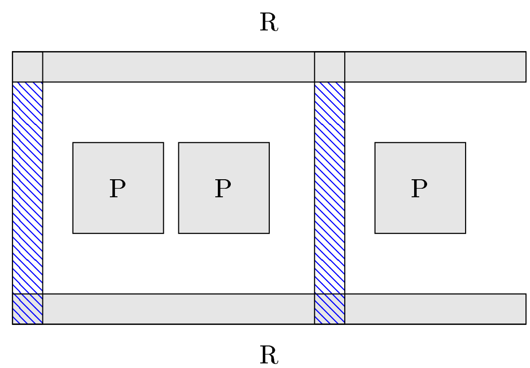


Figure 4.9: An illustration of the proof of Proposition 4.13. The result of the procedure is the colored rectangle.

Proof. 1. **Gluing multiple times the same block P horizontally:**

   Consider P some n-block in the language of X, and take 2^k copies of it. We group them by two and glue the pairs horizontally, at distance $f(n)$. Then glue the obtained patterns after grouping them by two, at distance $f(2n + f(n))$, and repeat this operation until having one rectangular block Q. The size of this block is equal to $(2id + f)^{\circ k}(n)$.

2. **Glue the same rectangular pattern over and under the obtained one:**

   Then consider some $(2id + f)^{\circ k}(n) \times (r - 1)$ pattern R, where r is the rank of the SFT X. Glue it on the top of Q with $f(\max((2id + f)^{\circ k}(n), n, r))$ lines between the two rectangles. Then glue the rectangle R under the obtained pattern with

$$f(\max(f(\max((2id + f)^{\circ k}(n), n, r)) + r + n, (2id + f)^{\circ k}(n)))$$

lines between them. For k great enough (depending on n), these two last distances are equal to $f((2id+f)^{\circ k}(n))$, and $f(f((2id+f)^{\circ k}(n))+n+r)$ respectively. By the gluing property, the obtained pattern (see Figure 4.9) is in the language of X.

3. **Pigeon hole principle on the columns of the obtained pattern:**

   Consider the $(r-1)\times(r+n+f((2id+f)^{\circ k}(n))+f(f((2id+f)^{\circ k}(n)+n+r))$ sub-patterns that appear on the bottom of the columns just on the right of each occurrence of the pattern P. There are 2^k of them, and there are at most $(|\mathcal{A}|)^{r+n+f((2id+f)^{\circ k}(n)+f(f((2id+f)^{\circ k}(n)+n+r))}$ different possibilities. From the fact that $f \leq id$, it follows that

   $$(|\mathcal{A}|)^{r+n+f((2id+f)^{\circ k}(n)+f(f((2id+f)^{\circ k}(n)+n+r))} \leq (|\mathcal{A}|)^{2(r+n)+2f(3^k n)} \leq 2^k$$

   for k great enough.

   By the pigeon hole principle, two of these patterns are equal. Consider the rectangle between these two occurrences (including the second one).

4. **Construction of a periodic configuration containing P:**

   This rectangle can be repeated on the whole plane to get a periodic configuration which is in X.

   The set of periodic configurations obtained by this method is dense in X (for every pattern in its language appear in such a configuration).

□

Proposition 4.14. *A subshift X verifying the conditions of Proposition 4.13 with f computable has decidable language.*

Proof. It is sufficient to prove that it is possible to decide if a block is in the language of X. Indeed, a pattern is in the language of X if and only if it is a sub-pattern of a block in the language of X whose size is determined by the pattern. In order to decide if a pattern is in the language of X, we consider all the blocks of this size having this pattern as sub-pattern, and use the algorithm on block to decide if these blocks are in the language of X. The pattern is in the language of X if and only if one of these blocks is in it.

We proved, in the proof of Proposition 4.13, that any n-block is sub-pattern of a rectangular pattern whose size is bounded by a computable function of n (since f is computable), whose topmost r lines are equal to the bottommost r lines and the leftmost r columns are equal to the rightmost r columns. To decide if some n-block P is in the language of X, we test for all the rectangular patterns that verify these conditions, if it has P as sub-pattern. The block P is in the language if and only if this is the case for at least one of these patterns. □

4.3.3 An example of linearly block gluing SFT with non decidable language

The property obtained in Proposition 4.4.2 is no longer true considering linearly block gluing subshifts of finite type. In this section we provide an example of linearly block gluing subshift of finite type having non decidable language:

Proposition 4.15. *There exists some $O(n)$-block gluing $\mathbb{Z}^2$-SFT with non decidable language.*

Idea of the proof: *we construct a structure subshift which consists of infinite and constantly growing areas for the computations of machines. These areas can be distorted by shifting its adjacent lines one with respect to the other. This can be done in any of the two possible directions. This allows the linear block gluing. Then we implement a universal Turing machine, and forbid it to stop. This implies the non decidability.*

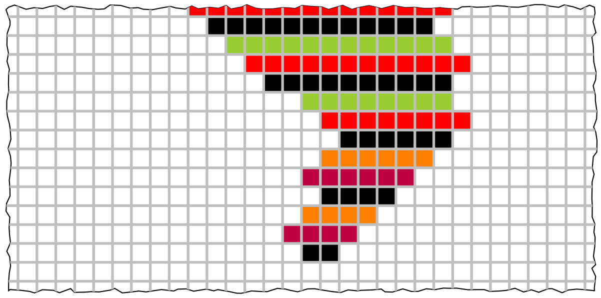


Figure 4.10: An example of configuration that respects the rules of the first layer of $X_{\texttt{Undec}}$.

Proof. Let X_{undec} be a subshift, product of two layers. Here is a description of these layers:

1. **Computation areas layer:**

   This layer has the following symbols:

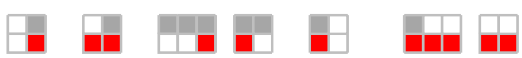
   Its local rules are the following ones:

   - two horizontally adjacent non blank symbols have the same color.
   - two vertically adjacent non blank symbols verify the following rules:
     (a) if the bottom symbol is , the top symbol is or .
     (b) if the bottom symbol is , the top symbol is .
     (c) if the bottom symbol is , the top symbol is .
     (d) if the bottom symbol is or , the top symbol is .

     These rules allow to shift a row of the area from the one under, choosing the direction. Moreover the shifts happen each time by groups of two, so that the shift counterbalances the growth of the area.
   - the patterns

     are forbidden, where the gray symbol stands for any non blank symbol.
   - the patterns

     are forbidden, where the gray symbol stands for any non blank symbol. Similar rules are imposed, replacing the red symbol with a green one.
   - the patterns

     are forbidden, where the gray symbol stands for any non blank symbol. Similar rules replacing the red symbol with an orange one. These rules allow to control the shape of the areas.

   These rules imply that:

- above $\square\,\blacksquare^n\,\square$ there is ${\color{red}\blacksquare}\,{\color{red}\blacksquare}^n\,{\color{red}\blacksquare}$, or ${\color{purple}\blacksquare}\,{\color{purple}\blacksquare}^n\,{\color{purple}\blacksquare}$.
- above $\square\,{\color{red}\blacksquare}^n\,\square$ there is ${\color{green}\blacksquare}^n\,\square\square$.
- above $\square\,{\color{purple}\blacksquare}^n\,\square$ there is $\square\square\,{\color{orange}\blacksquare}^n$.
- above $\square\,{\color{green}\blacksquare}^n\,\square$ there is $\blacksquare^n\,\square\square$.
- above $\square\,{\color{orange}\blacksquare}^n\,\square$ there is $\square\square\,\blacksquare^n$.

The computation areas consist of colored areas. They lie on a background of $\square$ symbols. These areas are distorted infinite triangles: in two adjacent rows, the intersection of the area with the top row is larger than in the bottom row by one position on the right and one position on the left. Then this row is shifted or not, horizontally in one of the two directions, depending on the colors of the rows. See Figure 4.10 for an illustration.

2. **Machines layer:**

   The second layer consists in the implementation of a Turing machine over these areas.

   The symbols are the elements of $\mathcal{Q} \times \mathcal{A} \times \{\rightarrow, \leftarrow\}$, and the local rules are the following ones:

   (a) the blank symbols are superimposed with a blank symbol, and the Turing machine symbols are superimposed over non blank symbols.

   (b) Moreover, considering a 3×2 pattern whose projection on the first layer is fully colored and the bottom row is black, the rules of the space-time diagram of the machine apply. When the bottom row is not black, the symbols of this row are copied on the top row, on the shifted position according to the color of the bottom row. There rules of the space time diagram are adapted when on the border of the area.

   (c) Any halting state is forbidden.

   These rules imply that for two adjacent rows of an area:

   - if the bottom row is colored black, then the top row is the image of the bottom row by the machine process.
   - in the other cases, the top row is just the image of the bottom row by the shift in the direction corresponding to the color of the bottom row.

   We use a universal Turing machine, which has the following behavior when the initial tape is written with $w\#^\infty$ for w a word on $\{0, 1\}$: it reads the word w which codes for the number of a Turing machine, and then simulates this machine on empty tape.

3. **Properties of this subshift:**

   (a) **The language of this subshift is not decidable:** if it was, we would be able to decide which of the words $w\#^n$ can be written on the bottom of an area. This is impossible since the halting problem is not decidable.

   (b) **This subshift is sharp linearly block gluing:** the worst case for gluing two blocks is when these blocks are filled with colored symbols in the first layer. In order to glue them, we complete the projection of the two blocks in the first layer, as in example 4.10, into the bottom of a computation area, surrounding it with blank symbols except on the top. Then we complete the trajectory of the machine head if there is one. The two extended patterns can be glued horizontally without constraint on the distance, because lines can be shifted towards opposite directions. For vertical gluing, we extend one of this patterns shifting the area in one direction so that the columns above this patterns are blank. The number of rows and columns depends linearly on the size of this pattern. Then we glue the second pattern on the top, shifting the area in the opposite direction and filling all the positions of $\mathbb{Z}^2$ left undefined by blank symbols.

□

4.3.4 Existence of aperiodic linearly block gluing subshifts

In this section, we give a proof of the following theorem:

Theorem 1.1. *There exists a linearly block gluing aperiodic $\mathbb{Z}^2$-SFT.*

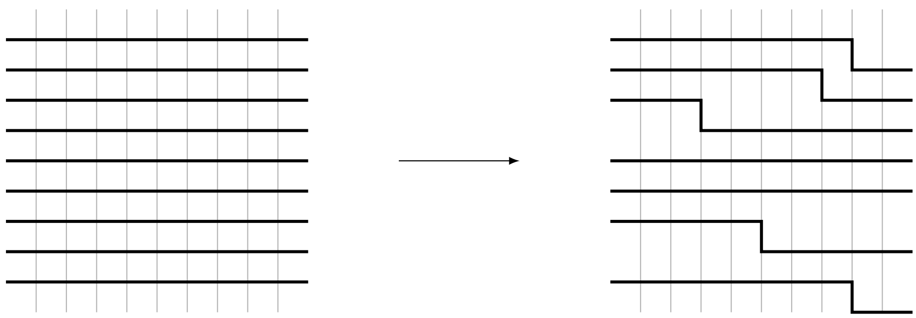


Figure 4.11: Illustration of the distortion principle of the transformation for the proof of Theorem 1.1.

Idea of the proof: *Let us recall the idea of the proof for Theorem 1.1, presented in the introduction. This proof uses an operator on subshifts which transforms linearly net gluing subshifts of finite type into linearly block gluing subshifts of finite type. Moreover, it preserves the aperiodicity. The principle of this transformation is to distort $\mathbb{Z}^2$, as illustrated on Figure 4.11, multiple times and in different directions.*

We then apply this transformation on the Robinson subshift which is known to be aperiodic and that we proved to be linearly net gluing (Proposition 4.11).

4.3.4.1 A subshift inducing pseudo-coverings by curves

Definition Let us denote Δ the $\mathbb{Z}^2$ SFT on alphabet $\{\rightarrow, \downarrow\}$, defined by the following forbidden patterns:

$$\begin{matrix} \downarrow \\ \downarrow \end{matrix}, \quad \begin{matrix} \rightarrow & \downarrow \\ \rightarrow & \rightarrow \end{matrix}.$$

Pseudo-coverings by curves Let us introduce some words in order to talk about the global behavior induced by these rules:

- An (infinite) **curve** in $\mathbb{Z}^2$ is a set $\mathcal{C} = \varphi(\mathbb{Z})$ for some application $\varphi : \mathbb{Z} \rightarrow \mathbb{Z}^2$ such that for all $k \in \mathbb{Z}$, $\varphi(k+1) = \varphi(k) + (1,0)$ or $\varphi(k+1) = \varphi(k) + (1,-1)$.
- We say that a curve is **shifted downwards** at position $\mathbf{j} \in \mathbb{Z}^2$ when there exists some $k \in \mathbb{Z}$ such that $\varphi(k) = \mathbf{j}$ and $\varphi(k+1) = \mathbf{j} + (1,-1)$.

- A **pseudo-covering of $\mathbb{Z}^2$ by curves** is a sequence of curves $(\mathcal{C}_k)_{k\in\mathbb{Z}}$ such that for every $\mathbf{j} \in \mathbb{Z}^2$, there exists some $k \in \mathbb{Z}$ such that $\{\mathbf{j}, \mathbf{j}+(0,1)\} \bigcap \mathcal{C}_k \neq \emptyset$ (meaning that every element of $\mathbb{Z}^2$ is in a curve or the vector just above is), and for every $k \neq k'$, $\mathcal{C}_k \bigcap \mathcal{C}_{k'} = \emptyset$ (the curves do not intersect). We say that two curves in this pseudo-covering are **contiguous** when the area delimited by these two curves does not contain any third curve. The gap between two contiguous curves in some column is the distance between the intersection of these two curves with the column. This gap is 0 or 1 between two contiguous curves in a pseudo-covering.

A configuration in Δ induces a pseudo-covering by curves Let $\delta \in \Delta$. Let us consider the pseudo-covering of $\mathbb{Z}^2$ by curves $(\mathcal{C}_k(\delta))_{k\in\mathbb{Z}}$, such that $\mathcal{C}_k(\delta) = \varphi_{\delta,k}(\mathbb{Z})$ and where $\varphi_{\delta,k}$ is as follows.

If $\delta_{(0,0)} =\rightarrow$, then $(0,0) = \varphi_{\delta,0}(0)$. Else $(0,1) = \varphi_{\delta,0}(0)$. In addition, for m the biinfinite sequence of integers such that $\delta_{(m_k,0)}$ is the kth $\rightarrow$ in the column 0, counting from the previous considered one, then $(m_k, 0) = \varphi_{\delta,k}(0)$.

For every $\mathbf{i} \in \mathbb{Z}^2$ such that $\mathbf{i} = \varphi_{\delta,k}(n)$ for some $k \in \mathbb{Z}, n \in \mathbb{Z}$:

- if $\delta_{\mathbf{i}} =\rightarrow$, and $\delta_{\mathbf{i}+(1,0)} =\downarrow$, then $\varphi_{\delta,k}(n+1) = \mathbf{i} + (1,-1)$ (the curve is shifted downwards in this column).
- else $\delta_{\mathbf{i}} =\rightarrow$ and $\delta_{\mathbf{i}+(1,0)} =\rightarrow$, then $\varphi_{\delta,k}(n+1) = \mathbf{i} + (1,0)$.

The first rule implies that all the curves of this pseudo-covering can not be shifted downwards multiple times in the same column. The second one implies that if a curve is shifted downwards at position $\mathbf{i} \in \mathbb{Z}^2$, then there is no curve going through position $\mathbf{i} - (1,1)$.

Figure 4.12 gives an illustration of the definition of a curve in a configuration of Δ.

$$\begin{array}{cccc} \rightarrow & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \downarrow & \downarrow & \downarrow \\ \downarrow & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow \end{array} \in \mathcal{L}(\Delta) \longmapsto$$

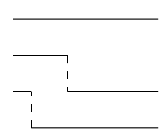


Figure 4.12: An example of an admissible pattern in Δ and the curves going through it.

4.3.4.2 Distortion operators on subshifts of finite type

Let $\mathcal{A}$ be some alphabet. Denote $S_{\mathcal{A}}$ the set of SFT over $\mathcal{A}$. We introduce operators $d_{\mathcal{A}} : \mathcal{S}_{\mathcal{A}} \rightarrow \mathcal{S}_{\tilde{\mathcal{A}}}$, with $\tilde{\mathcal{A}} = \left(\mathcal{A} \cup \{\square\}\right) \times \{\rightarrow, \downarrow\}$.

Pseudo-projection Consider the subshift $\Delta_{\mathcal{A}} \subset \left(\mathcal{A} \cup \{\square\}\right)^{\mathbb{Z}^2} \times \Delta$, where the forbidden patterns are the ones defining Δ and the patterns where a symbol in $\mathcal{A}$ is superimposed to a $\downarrow$ symbol or where $\square$ is superimposed to a $\rightarrow$ symbol.

Define a **pseudo-projection** $\mathcal{P} : \Delta_{\mathcal{A}} \rightarrow \mathcal{A}^{\mathbb{Z}^2}$, as follows: for $(y, \delta) \in \Delta_{\mathcal{A}}$,

$$(\mathcal{P}(y,\delta))_{i,j} = y_{\varphi_{\delta,j}(i)}.$$

Notice that the function $\mathcal{P}$ is continuous but not shift invariant.

We denote π_1 the projection on the first layer ($\pi_1(y, \delta) = y$), and π_2 the projection on the second layer.

Definition of the operators Let X be some SFT on the alphabet $\mathcal{A}$, and define $d_{\mathcal{A}}(X) = \mathcal{P}^{-1}(X)$. Denoting r the rank of the SFT X, $d_{\mathcal{A}}(X)$ can be defined by imposing that, considering the intersection of a set of r contiguous curves with r consecutive columns, the corresponding r-block is not a forbidden pattern in X. Because the gap between two contiguous curves is bounded, $d_{\mathcal{A}}(X)$ is defined by a finite set of forbidden patterns. Then this is a SFT.

One can think to $d_{\mathcal{A}}(X)$ as having two layers. The first one has alphabet $\left(\mathcal{A} \cup \{\square\}\right)$ and called the X layer. The second one has alphabet $\{\rightarrow, \downarrow\}$ and is called the Δ layer. For X is the subshift X_R or X_{adR}, Figure 4.13 shows an example of pattern in the X layer of the subshift $d_{\mathcal{A}}(X)$ whose pseudo-projection is the supertile south west order two supertile.

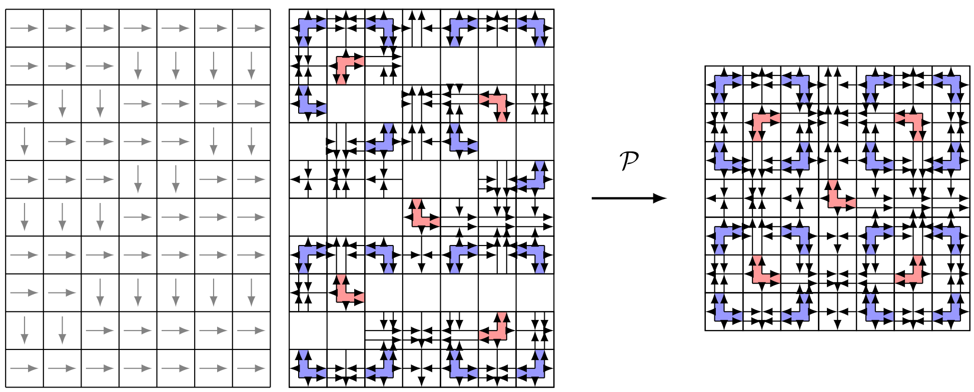


Figure 4.13: Example of a pattern which is sent to the south west two order supertile by pseudo-projection.

Properties of the operators $d_{\mathcal{A}}$ We use the following properties of the operators $d_{\mathcal{A}}$ in order to prove Theorem 1.1.

Proposition 4.16. *For an aperiodic SFT X on the alphabet $\mathcal{A}$, $d_{\mathcal{A}}(X)$ is also aperiodic.*

Idea: *the main argument of this proof is that if a configuration in $d_{\mathcal{A}}(X)$ is periodic, then the projection on the Δ layer is periodic. This means that although there is a distortion of the configuration in X, the distortion is done in a periodic way. From this, we deduce that the pseudo-projection on X of this configuration is periodic.*

Proof. Assume that there exists a configuration $z \in d_{\mathcal{A}}(X)$ which is periodic: there exists $n > 0$ such that for all i, j, $z_{i+n,j} = z_{i,j+n} = z_{i,j}$. We will prove that the pseudo-projection of z on X, $x = \mathcal{P}(z)$ is periodic.

1. **Coding the positions of intersections of the curves with a column:**

   To each column k in z we associate the bi-infinite word ω^k in $(\mathbb{Z}/n\mathbb{Z})^{\mathbb{Z}}$ such that for all $i \in \mathbb{Z}$, ω_i^k is the element $\overline{m_i}$ of $\mathbb{Z}/n\mathbb{Z}$, class modulo n of m_i where $(0, m_i)$ is the intersection position of the ith curve of $\pi_2(z)$ with the column k.

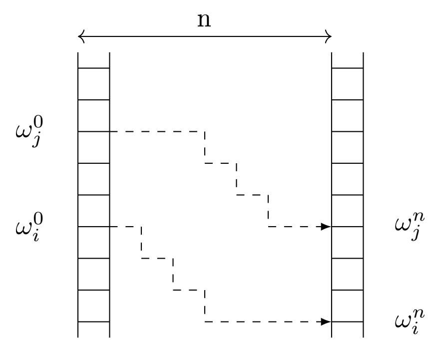


Figure 4.14: Schema of the proof of Proposition 4.16.

2. **Function relating the codings of two columns:**

   Following a curve (see Figure 4.14) from the column 0 to the column n, we get an application ψ from the set of possible $\overline{m_i}$ into itself. This comes from the vertical periodicity of the projection of z on the second layer. The word ω^n is obtained from ω^0 applying ψ to all the letters in ω^n.

3. **Coding of the intersections and periodicity:**

   Since ψ is an invertible function from a finite set into itself (indeed, we have an inverse map following the curve backwards), there exists some $c > 0$ integer such that $\psi^c = \text{Id}$. As a consequence, $\omega^{nc+j} = \omega^j$ for all integers j. That means that the column $cn + j$ is obtained by shifting kn times downwards the column j, for some $k \geq 0$. Using the horizontal periodicity of y, we then have that

   $$(x_{j,l})_{l\in\mathbb{Z}} = (x_{cn+j,l+kn})_{l\in\mathbb{Z}}.$$

   Using the vertical periodicity, that $(x_{j,l})_{l\in\mathbb{Z}} = (x_{cn+j,l})_{l\in\mathbb{Z}}$, hence the configuration $x \in X$ is periodic, which can not be true.

As a consequence, no configuration in $d_{\mathcal{A}}(X)$ can be periodic. Thus this subshift is aperiodic. □

The following proposition will be a useful tool in order to prove that the operators $d_{\mathcal{A}}$ transform linearly net gluing subshifts into block gluing ones.

Proposition 4.17 (Completing blocks)**.** *There exists an algorithm $\mathcal{T}$ that, taking as input some locally admissible n-block p of Δ, outputs a rectangular pattern $\mathcal{T}(p)$ which has p as a sub-pattern and such that:*

- *the number of curves in $\mathcal{T}(p)$ is equal to the number of its columns,*
- *the dimensions of $\mathcal{T}(p)$ are smaller than $5n$,*
- *the top and bottom rows of $\mathcal{T}(p)$ have only $\rightarrow$ symbols- this means that all the curves crossing $\mathcal{T}(p)$ comes from its left side and go to the right side.*

Remark 27. *The properties of the pattern $\mathcal{T}(p)$ ensure that this is a globally admissible pattern. Hence every locally admissible pattern of the subshift Δ is globally admissible.*

Idea: *the proof consists in extending the curves that cross a pattern from above and below. Then we add curves on the top and bottom that are straighter and straighter.*

Proof. If p is a 1-block, and p is a single $\rightarrow$, then the result is direct. If p is a single $\downarrow$, then it can be extended in

$$\begin{matrix} \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \downarrow & \downarrow \\ \downarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow \end{matrix},$$

which verifies the previous assertion.

If p is a n-block with $n \geq 2$:

First step. Extending the curves that enter in the block upside/downside:

1. If in the top row of the block p there is the pattern $\downarrow\rightarrow$ - this means that there is an incoming curve (the position of the $\rightarrow$ is in this curve) - then we add symbols in the row just above. We extend each of the incoming curves, considering the curves from left to right, in this row. For this purpose we add a $\downarrow$ over the $\rightarrow$ for each of the patterns $\downarrow\rightarrow$. Then we add $\rightarrow$ symbols on the left of this one until meeting another $\downarrow$ or the left side of the block. If in the added row there are $\downarrow\rightarrow$ patterns, then return to the beginning of this step. Else, stop.
2. Do similar operations on the bottom of the block.

Since the number of $\downarrow\rightarrow$ patterns in the top row is strictly decreasing, this series of operations stops at some point.

Example 4.18. *If we take p the following 4-block*

$$\begin{matrix} \rightarrow & \rightarrow & \downarrow & \rightarrow \\ \rightarrow & \downarrow & \rightarrow & \rightarrow \\ \downarrow & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow \end{matrix}$$

at this point, we obtain:

$$\begin{matrix} \rightarrow & \rightarrow & \rightarrow & \downarrow \\ \rightarrow & \rightarrow & \downarrow & \rightarrow \\ \rightarrow & \downarrow & \rightarrow & \rightarrow \\ \downarrow & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow \end{matrix}$$

Second step. Completing the pattern on the top and bottom until the top row and bottom row are straight:

While the top curve of the pattern is not straight, apply the following procedure:

1. On the top of the last column, add a $\downarrow$ and keep adding $\downarrow$ on the left until meeting on the left an already defined symbol (there can be such symbols, introduced in the first step) or the left extremity.
2. Add another curve above by the following procedure. Add a $\rightarrow$ on the top of the last column, and then add $\rightarrow$ symbols on the left until meeting an already defined symbol on the left. When that happens, add a $\downarrow$ above and then add $\rightarrow$'s on the left until reaching a defined symbol. Repeat this operation until reaching the first column.

Since the number of times that the top curve is shifted downwards decreases at each step, this series of operations stops.

Do similar operations on the bottom.

Example 4.19. *At this point, we obtain:*

$$\begin{array}{cccc}
\rightarrow & \rightarrow & \rightarrow & \rightarrow \\
\rightarrow & \rightarrow & \rightarrow & \downarrow \\
\rightarrow & \rightarrow & \downarrow & \rightarrow \\
\rightarrow & \downarrow & \rightarrow & \rightarrow \\
\downarrow & \rightarrow & \rightarrow & \rightarrow \\
\rightarrow & \rightarrow & \rightarrow & \rightarrow
\end{array}$$

Third step. Equalization of the number of curves and the number of columns:

If the number of columns is smaller than the number of curves, then add a number of columns equal to the difference, by adding copies of the last column on its right side. If the number of curves is smaller, then add lines of $\rightarrow$ symbols on the top.

Example 4.20. *After this last step we obtain:*

$$\begin{array}{ccccc}
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \\
\rightarrow & \rightarrow & \rightarrow & \downarrow & \downarrow \\
\rightarrow & \rightarrow & \downarrow & \rightarrow & \rightarrow \\
\rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow \\
\downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \\
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow
\end{array}$$

For p some n-block, the dimensions of $\mathcal{T}(p)$ are smaller than the sum of:

1. the dimension of p (equal to n)
2. two times the number of entering curves by the top and outgoing by the bottom (one for completing the curves (first step)
3. and one for reducing the shifts (second step)).

Each one of these numbers is smaller than n. The third step does not make this bound greater, because in this pattern the number of curves is smaller than the number of lines. As a consequence, the dimensions of $\mathcal{T}(p)$ are smaller than $5n$. □

Proposition 4.21. *Let X be some linearly net gluing subshift on alphabet $\mathcal{A}$. There exists some vector function $\boldsymbol{u}'$ and a function g that:*

1. *take as arguments two n-blocks p, q in the language of X for some n,*
2. *respectively associates to these blocks an element of $\mathbb{Z}^2$ and an element of $\mathbb{N}$*

These functions verify that for any pair of n-blocks p, q in the language of $d_{\mathcal{A}}(X)$, the gluing set of p relative to q contains infinite columns regularly displayed. Moreover, the gluing set contains regularly displayed positions in a central column:

$$\boxed{g(p,q)(\mathbb{Z}\backslash\{0\})\boldsymbol{e}^2 + \boldsymbol{u}'(p,q) \subset \Delta_{d_{\mathcal{A}}(X)}(p,q)},$$

$$\boxed{g(p,q)(\mathbb{Z}\backslash\{-3,...,3\})\boldsymbol{e}^1 + \mathbb{Z}\boldsymbol{e}^2 + \boldsymbol{u}'(p,q) \subset \Delta_{d_{\mathcal{A}}(X)}(p,q)},$$

where function g verifies that

$$\max_{p,q} g(p,q) = O(n),$$

where the maximum is over the n-blocks.

Idea: *this proof consists in analyzing how the operator acts on the gluing set of a n-block p relative to another pattern q. The operator allows perturbations to be introduced on these sets.*

Proof. Let X be some f net gluing subshift on alphabet $\mathcal{A}$, where $f(n) = O(n)$.

Formulation of the linear net gluing of X: This means that there exist two function $\mathbf{u}$: $\mathcal{L}_n(X)^2 \to \mathbb{Z}^2$ and $\tilde{f} : \mathcal{L}_n(X)^2 \to \mathbb{N}$ such that for all $n > 0$ and every pair of n-blocks r, s in the language of X, the gluing set of r relative to s in X contains

$$\mathbf{u}(r,s) + (n + \tilde{f}(r,s))(\mathbb{Z}^2 - (0,0)),$$

and for all r, s n-blocks,

$$\tilde{f}(r,s) \leq f(n).$$

We consider in this proof that $\mathbf{u} = \mathbf{0}$, since this proof can be adapted to a general function $\mathbf{u}$ without difficulty.

Sufficient conditions to verify: It is sufficient to prove the statement of the proposition for patterns whose projection on Δ are $\mathcal{T}(\pi_2(p))$ and $\mathcal{T}(\pi_2(q))$. Indeed, the size of these patterns is bounded by a linear function of the size of the patterns $p, q \in d_{\mathcal{A}}(X)$.

Let p, q two n-blocks in the language of the subshift $d_{\mathcal{A}}(X)$. Without loss of generality, we can consider that the patterns $\mathcal{T}(\pi_2(q))$ and $\mathcal{T}(\pi_2(p))$ have the same number of curves crossing them. We denote $\tilde{p}$ and $\tilde{q}$ some admissible patterns whose projections of Δ are respectively $\mathcal{T}(\pi_2(p))$ and $\mathcal{T}(\pi_2(q))$. The pseudo-projections of these patterns on X are m-block of X, where m is the number of curves in $\mathcal{T}(\pi_2(p))$ and $\mathcal{T}(\pi_2(q))$. This is due to the fact that the top and bottom rows of these patterns are straight. These pseudo-projections are denoted $\mathcal{P}(\tilde{p})$ and $\mathcal{P}(\tilde{q})$, according to previous notations.

We place the pattern $\mathcal{P}(\tilde{q})$ on position $(0,0)$.

The gluing sets of $d_{\mathcal{A}}(X)$ contain infinite columns periodically displayed: Let us show that the gluing set of $\tilde{p}$ relative to $\tilde{q}$ contains infinite columns periodically displayed. This corresponds the second inclusion in the statement of the proposition.

Let $k \geq 4$ and l integers such that $l \neq 0$, and consider some vector

$$\mathbf{u} = (m + \tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q}))(k,l).$$

We will prove the following:

- When $l \geq 2$ and t is any integer such that

$$0 \leq t \leq \tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q})) + m,$$

  the position $\mathbf{u} + t.\mathbf{e}^2$ is in the gluing set in $d_{\mathcal{A}}(X)$ of the pattern $\tilde{p}$ relative to $\tilde{q}$.

- When $l \leq -2$, this set contains the position $\mathbf{u} - t.\mathbf{e}^2$, for the same integers t.

- When $l = 1$ or $l = -1$, then this set contains $\mathbf{u} - t.\mathbf{e}^2$, for all t such that

$$0 \leq t \leq 2(\tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q})) + m).$$

As a consequence, this gluing set contains the whole infinite column that contains

$$(m + \tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q}))(k, 0)$$

for all $k \geq 8$. Indeed, it contains an infinity of segments which overlap only on their border. We have the same property for $k \leq -4$, by reversing p and q.

1. **When $|l| \geq 2$:**

   Let t be some integer such that

$$0 \leq t \leq \tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q})) + m.$$

   In the case $l \geq 2$, we do the following operations. See a schema on Figure 4.15.

   (a) We extend the curves crossing the pattern $\tilde{q}$ in a straight way until infinity.

   (b) In the case $l \geq 2$, we add straight curves below the obtained pattern. We do that in such a way that these curves have gap 0 between them. On the top, we introduce t times a straight infinite curve with gap 1 with the curve below. Then we add

$$l(\tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q})) + m) - t - m$$

   straight infinite curves with gap 0 with the curve below.

   (c) Then we add the pattern $\tilde{p}$ on position $\mathbf{u} + t.\mathbf{e}^2$.

   (d) We extend the curves crossing this pattern in a straight way until infinity.

   (e) We add straight lines on the top, without gaps between them.

   (f) Then we color the curves with elements of the curves with elements of $\mathcal{A}$ such that the configuration is admissible. This is possible from the net gluing property of the subshift X. Indeed, the position of the pattern $\mathcal{P}(\tilde{p})$ relatively to $\mathcal{P}(\tilde{q})$ in the pseudo-projection of this configuration is $\mathbf{u}$. Moreover, this vector is in the gluing set of the first pattern relatively to the second one.

   The case $l \leq -2$ is similar. The difference is that the pattern $\tilde{p}$ appears on position $\mathbf{u} - t.\mathbf{e}^2$.

2. **When $l = 1$ or -1:**

   Here we prove that the pattern $\tilde{p}$ can be glued relatively to $\tilde{q}$ on position $\mathbf{u} - t.\mathbf{e}^2$.

   The steps of a construction of a configuration that supports this gluing are as follows:

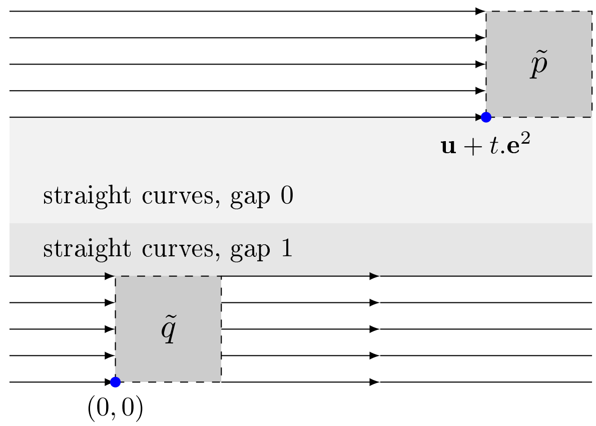


Figure 4.15: Illustration of the construction for the proof of Theorem 4.26 when $l \geq 2$.

(a) **Compactification of the outgoing curves:**

We extend $\mathcal{T}(\pi_2(q))$ using the following procedure. While in the last column of the pattern there is some sub-pattern $\begin{smallmatrix}\rightarrow\\ \downarrow\end{smallmatrix}$ (meaning that there is a gap between two outgoing curves), do the following: on the right of the patterns $\begin{smallmatrix}\rightarrow\\ \downarrow\end{smallmatrix}$, write $\begin{smallmatrix}\downarrow\\ \rightarrow\end{smallmatrix}$ and write a copy of the other $\rightarrow$ symbols on their right side.

Example 4.22. *Taking the same example as in the proof of Proposition 4.17, the result is:*

$$\begin{array}{ccccccc}
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow \\
\rightarrow & \rightarrow & \rightarrow & \downarrow & \downarrow & \downarrow & \rightarrow \\
\rightarrow & \rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \\
\rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \\
\downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \\
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow
\end{array}$$

Since there are m curves the number of additional columns on the right for this step is smaller than m. Indeed, one column is sufficient to reduce the gap between a curve and the curve just below.

(b) **Making the curves shift:**

We add columns on the right of the extension of $\mathcal{T}(\pi_2(q))$. We follow the following procedure, in order to make all the curves in it shift t times:

i. Consider the right part of the pattern constituted with $\rightarrow$ symbols and add a triangle made of $\rightarrow$ symbols except on the diagonal part. On this part we write $\downarrow$ symbols (this is the first shift).

Example 4.23. *Taking the same example as previously, the result is:*

$$\begin{array}{cccccccccccc}
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & & & & & \\
\rightarrow & \rightarrow & \rightarrow & \downarrow & \downarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow \\
\rightarrow & \rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \\
\rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & & \\
\downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & & & \\
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & & & &
\end{array}$$

ii. Then repeat $t-1$ times the following operation: add under each $\downarrow$ on the right side a $\rightarrow$ under, and after that a $\downarrow$ on the right of the $\rightarrow$.

Example 4.24. *Taking the same example as previously, with $t = 3$, the result is:*

$$\begin{array}{cccccccccccccc}
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & & & & & & & \\
\rightarrow & \rightarrow & \rightarrow & \downarrow & \downarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & & \\
\rightarrow & \rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \\
\rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow & \downarrow \\
\downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow \\
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow & \\
 & & & & & & & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow & & \\
 & & & & & & & & \rightarrow & \downarrow & \rightarrow & & & \\
 & & & & & & & & & \rightarrow & & & &
\end{array}$$

There are at most $2(m+\tilde{f}(\mathcal{P}(\tilde{p},\tilde{q})))$ additional columns for this step. Indeed, t is smaller than this number and only t columns are needed to shift a compact set of curves.

iii. Complete the curves with $\rightarrow$ symbols so that they end in the last column added.

Example 4.25. *Taking the same example as previously, with $t = 3$, the result is:*

$$\begin{array}{cccccccccccccc}
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & & & & & & & \\
\rightarrow & \rightarrow & \rightarrow & \downarrow & \downarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & & \\
\rightarrow & \rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \\
\rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow & \downarrow \\
\downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow \\
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow & \rightarrow \\
 & & & & & & & \rightarrow & \downarrow & \rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow \\
 & & & & & & & & \rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \\
 & & & & & & & & & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow
\end{array}$$

Here, no column is added.

iv. Then extend the curves straightly on a number of columns so that the total number of additional columns is equal to $\mathbf{u}_1 - m$. This is possible since the number of added columns at this point is smaller than $m + 2(m + \tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q})) \leq \mathbf{u}_1$, since $k \geq 4$.

(c) **Extension:**

Then, we extend these curves straightly until infinity on the east side and on the west side.

(d) **Additional curves:**

We add $\tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q}))$ curves on the top when $l = 1$ (resp. on the bottom when $l = -1$) of the obtained pattern at this point, without gap between them. This means that, from left to right, when the curve just below is shifted downwards the curve is also shifted immediately after.

(e) **Positioning the pattern $\mathcal{T}(\pi_2(p))$:**

   i. In the last columns where symbols were added in the last step, we position the pattern $\mathcal{T}(\pi_2(p))$. We place it on the top when $l = 1$ (resp. on the bottom when $l = -1$) of the last added curves.

   ii. After this we extend the curves on the west side straightly until infinity.

   iii. On the east side, we extend the curve without introducing gaps. This step is possible since the minimal value of k is taken sufficiently large. This means that the shifts of the outgoing curves of $\tilde{q}$ do not affect the area where the pattern $\tilde{p}$ is supposed to be glued.

   iv. On the top and bottom of the obtained pattern we add curves without introducing any gap in such a way that we fill $\mathbb{Z}^2$.

   v. In the end we add $\mathcal{A}$ symbols over the curves, when not already determined. This is possible from the linear net gluing property of X. Indeed, there are $\tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q}))$ added between the patterns $\tilde{p}$ and $\tilde{q}$ and the pattern $\tilde{p}$ is on a position in the column containing $\mathbf{u}$.

   See Figure 4.16 for an illustration of this case, when $l = 1$.

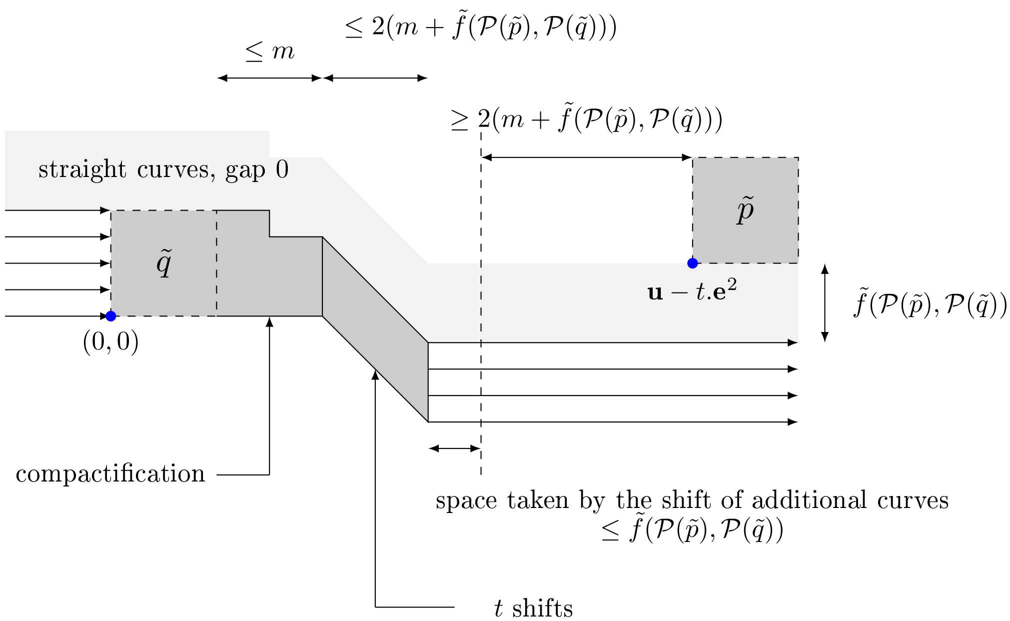


Figure 4.16: schema of the construction for the proof of Theorem 4.26 when $l = 1$.

From this construction we deduce that

$$\left\{\mathbf{w} \in \mathbb{Z}^2 \mid \mathbf{w}_1 = \mathbf{v}_1(g(p,q)),\ |\mathbf{v}_1| \geq 4\right\} \subset \Delta_{d_{\mathcal{A}(X)}}(p,q),$$

where

$$g(p,q) = m + \tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q})).$$

Thus,

$$\max_{p,q} g(p,q) = O(n).$$

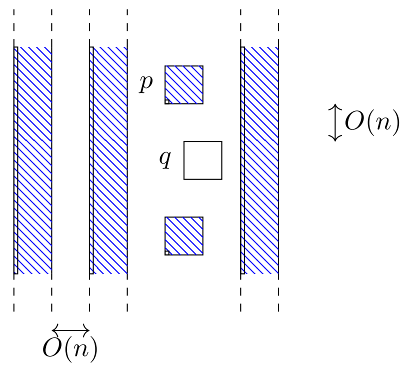


Figure 4.17: Schematic representation of a set of positions included in the gluing set of some pair of n-blocks in $d_{\mathcal{A}}(X)$.

Indeed, because $m \leq 5n$ and f is non decreasing, this number is smaller than $4(f(5n)+5n) = O(n)$. This comes from the fact that $f(n) = O(n)$.

The gluing sets of $d_{\mathcal{A}}(X)$ contain periodic positions in the central column: Consider some $\mathbf{u} = (m+\tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q})))\mathbf{v}$ with $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2)$, $\mathbf{v}_2 \neq 0$, and $\mathbf{v}_1 = 0$. The pattern p can be glued relatively to q in $d_{\mathcal{A}}(X)$ in position $\mathbf{u}$ in a configuration $x^{\mathbf{u}}$.

Indeed, the pattern $\mathcal{T}(\pi_2(q))$ can be glued relatively to $\mathcal{T}(\pi_2(p))$ in Δ with relative position $\mathbf{u}$. In order to prove that one can glue the two patterns with this relative position. Then one completes straightly the curves that go through the two patterns and fulfill $\mathbb{Z}^2$ with straight curves. One shifts the configuration in such a way that $\pi_2(q)$ appears in position $(0,0)$. Then one completes this configuration with letters in $\mathcal{A}$ so that the pseudo-projection is $x^{\mathbf{u}}$.

This means that

$$\left\{\mathbf{w} \in \mathbb{Z}^2 \mid \mathbf{w}_2 \in (m+\tilde{f}(\mathcal{P}(\tilde{p}), \mathcal{P}(\tilde{q})))(\mathbb{Z}\backslash\{0\}),\ \mathbf{w}_1 = 0\right\} \subset \Delta_{d_{\mathcal{A}}(X)}(p,q).$$

The Figure 4.17 shows the set of positions that we proved to be in the gluing set of the pattern p relatively to q.

$\square$

Let us denote ρ the transformation on subshifts that acts as a rotation by an angle $\pi/2$. Let X be a subshift on an alphabet $\mathcal{A}$ and defined by a set $\mathcal{F}$ of forbidden patterns. Then $\rho(X)$ is the subshift on alphabet $\mathcal{A}$ defined by the set of patterns that are image by rotation of the patterns in $\mathcal{F}$. Thus ρ transforms SFT into SFT.

Theorem 4.26. *The operator $d_{\tilde{\mathcal{A}}} \circ \rho \circ d_{\mathcal{A}}$ transforms linear net-gluing subshifts of finite type into linear block gluing ones.*

Idea: *the idea is to see how this operator acts on the gluing sets of n-blocks p relative to another q and see that it fulfills the part of $\mathbb{Z}^2$ outside of a box containing q whose size is $O(n)$.*

Proof. Since ρ acts as a $\pi/2$ rotation over patterns (and thus on configurations), the gluing set of some n-block p in the language of $\rho \circ d_{\mathcal{A}}(X)$ relatively to another one q contains the positions shown by

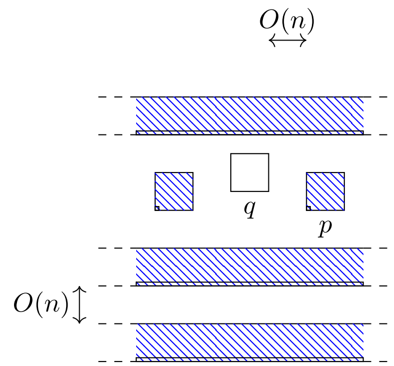


Figure 4.18: Schematic representation of a set of positions included in the gluing set of some pair of n-blocks in $\rho \circ d_{\mathcal{A}}(X)$.

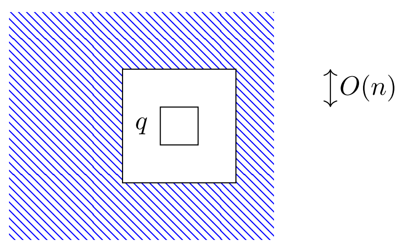


Figure 4.19: Schematic representation of a set of positions included in the gluing set of some pair of n-blocks in $d_{\tilde{\mathcal{A}}} \circ \rho \circ d_{\mathcal{A}}(X)$.

Figure 4.18. Using the same procedure as in the proof of Proposition 4.21, we get that the gluing set of two n-blocks in the language of $d_{\tilde{\mathcal{A}}} \circ \rho \circ d_{\mathcal{A}}(X)$ contains some set of positions as in Figure 4.19. Indeed, this procedure introduced a vertical perturbation in the positions of the gluing sets. From the form of the sets included in the gluing sets on Figure 4.18, this perturbation transforms the gluing sets of the subshift $d_{\tilde{\mathcal{A}}} \circ \rho \circ d_{\mathcal{A}}(X)$ by fulfilling the plane outside the box having size $O(n)$.

This means that the subshift $d_{\tilde{\mathcal{A}}} \circ \rho \circ d_{\mathcal{A}}(X)$ is linearly block gluing. □

Proof. of Theorem 1.1: We know that X_{adR} is linearly net gluing. As a consequence, the subshift $d_{\tilde{\mathcal{A}}} \circ \rho \circ d_{\mathcal{A}}(X_{adR})$ is linearly block gluing. It is also aperiodic, since X_{adR} is aperiodic. □

4.4 Entropy of block gluing $\mathbb{Z}^2$-SFTs

In this section we present some results about the computability of entropy of block gluing $\mathbb{Z}^2$-SFT. The reader will find the proof of the main theorem in the next section.

4.4.1 Characterization of the entropies of $\mathbb{Z}^2$-SFT

Let us recall that $N_n(X) = |\mathcal{L}_n(X)|$ denotes the number of n-blocks in the language of a subshift X. Its entropy is defined as:

$$h(X) = \inf_n \frac{\log_2(N_n(X))}{n^2}$$

4.4.2 Computability of the entropy for the sub-logarithmic regime

In this section we show that the entropy of block gluing SFT with a sub-logarithmic gap function is computable. This generalizes Proposition 3.3 in [PS15]. This proposition states that a multidimensional SFT which is block gluing with a constant gap function is computable.

Proposition 4.27. *Let X be a SFT which is f block gluing. If the function f is such that $f(n) \in o(log(n))$ and $f \leq id$, then the $h(X)$ is computable.*

Proof. From Proposition , we get that the language of such a subshift is decidable. As a consequence, the sequence $(N_n(X))_n$ is computable. Together with Theorem 5.9, since the logarithm function verifies the hypotheses of this theorem, this implies that $h(X)$ is computable. □

Remark 28. *Let us note that although this theorem is used here, Theorem 5.9 is presented it in Chapter 5 for the reason that it relies on the hypothesis of decidability of the subshift.*

4.4.3 Characterization of the entropies of linearly block gluing $\mathbb{Z}^2$-SFT

In this section, we prove the following theorem:

Theorem 1.2. *The possible entropies of linearly block gluing $\mathbb{Z}^2$-SFT are exactly the non-negative Π_1-computable numbers.*

Corollary 4.28. *The possible entropies of transitive $\mathbb{Z}^2$-SFT are exactly the non-negative Π_1-computable numbers.*

4.4.3.1 Outline of the proof of Theorem 1.2

The steps list of the proof of Theorem 1.2 is the following:

1. We first prove that any Π_1-computable number h is the entropy of a linearly net gluing $\mathbb{Z}^2$-SFT. It is sufficient to realize the numbers in $[0,1]$. Indeed, in order to realize the numbers in $]1,+\infty[$ one can take the product of a linearly net gluing SFT having entropy in $[0,1]$ with a full shift. When h is in $\{0,1/4,1/2,3/4,1\}$, this is easy to find a linearly net gluing $\mathbb{Z}^2$-SFT whose entropy is h. Indeed, this can be done by allowing random bits on a regularly displayed set of positions in $\mathbb{Z}^2$. Hence we only have to realize the numbers in $[0,1]\backslash\{0,1/4,1/2,3/4,1\}$.

   The steps of the realization for these numbers are as follows:

   (a) First, for any Π_1-computable sequence $s \in \{0,1\}^{\mathbb{N}}$ and any $N > 0$, we construct in Section 4.5 a linearly net gluing $\mathbb{Z}^2$-SFT $X_{s,N}$. The integer N corresponds to a threshold and the sequence s corresponds to a control on the entropy of the subshift.

   This part is an adaptation of the construction in [HM10]. This construction uses an implementation of Turing machines whose work is to control the frequency of apparition of some random bits. These bits generate the entropy through frequency bits specifying if a random bit can be present or not. In this construction, the obstacles for transitivity (and thus linear net gluing) are the following ones:

      i. the identification of frequency bits over areas that are not closely related to the structures

      ii. and the possibility of degenerated behaviors of the Turing machines dynamics in infinite areas.

The first obstacle is solved by a modification of the identification areas (this mechanism is described in Section 4.5.3 and abstracted on Figure 4.20) corresponding to the insides of Robinson's structures defining computation units (they will be defined in a more precise way in the following). The second one is solved by the simulation of degenerated behaviors in all the computation units aside the intended behaviors. In order to ensure that the results of this simulation is not taken into account - this means that the frequency bits are not affected - we use error signals propagating through the border of Robinson's structures (this is described in Section 4.5.4 and Section 4.5.7, and abstracted on Figure 4.21). The subshift is net gluing since any pattern can be completed, with control on the size, into a pattern over a simulation area.

(b) In Section 4.6 we prove an explicit formula for the entropy of the subshifts $X_{s,N}$ which depends on the parameters s, N. Afterwards, for any $h \in [0,1]\backslash\{0, 1/4, 1/2, 3/4, 1\}$, we associate some $\mathbb{Z}^2$-SFT X_h by choosing some parameters s, N so that the entropy of X_h is h. Then we prove in Section 4.7 that this subshift is linearly net gluing.

2. We present in Section 4.8 adaptations $d_{\mathcal{A}}^{(r)}$, for all $r \geq 1$, of the operator $d_{\mathcal{A}}$ presented in Section 4.3.4.1. These operators verify the equalities

$$h(d_{\mathcal{A}}^{(r)}(Z)) = \frac{\log_2(1+r)}{r} + h(Z)$$

for any subshift Z. Since the additional entropy $\frac{\log_2(1+r)}{r}$ induced by the operator is computable and that the operator $d_{\mathcal{A}}^{(r)} \circ \rho \circ d_{\mathcal{A}}^{(r)}$ transforms linearly net gluing SFT into linearly block gluing ones, this allows any Π_1-computable number in $]0,1]$ to be realized as the entropy of a block gluing SFT. Since the entropy zero is trivial to realize, this means that any Π_1-computable number in $[0,1]$ is the entropy of a linearly block gluing SFT.

4.4.3.2 Description of the layers in the construction of the subshifts $X_{s,N}$

Here is a more precise description of the construction of the subshifts $X_{s,N}$.

First, we fix some parameters s and N, where s is a Π_1-computable sequence of $\{0,1\}^{\mathbb{N}}$, and $N > 0$ is an integer.

The construction of the subshift $X_{s,N}$ involves a hierarchy of computing units that we call cells. Each cell is divided into four parts and in each of these parts a machine works. The center (called nucleus) of each cell contains an information which codes for the behavior of the machines. Two of these machines compute and the other two simulate degenerated behaviors. This means in particular that the initial tape of the machine is left free and that machine heads can enter at any time and in any state on the two sides of the machine area. The aim of the computing machines is to control some random bits that generate the entropy, through frequency. These bits are grouped in infinite sets having frequency given by a formula. This allows the entropy to be expressed as the sum of three entropies $h_{\texttt{int}}$, $h_{\texttt{comp}}(s)$ and $h_{\texttt{sim}}(N)$. The first one is generated by random bits, and serve to place the entropy of the subshift in one of the quarters of the segment $[0,1]$. Each of the two other entropies is the sum of a series. The first one involves the sequence s and is due to random bits. The second one is generated by the symbols left free in the simulation area. We choose N so that the sum of $h_{\texttt{int}}$ and $h_{\texttt{sim}}(N)$ is smaller than $h_{\texttt{comp}}(s)$. By the choice of the sequence s we control the series whose sum is $h_{\texttt{comp}}(s)$. We choose this sequence so that the total entropy is h.

Here is a detailed description of the layers in this construction.

- *Structure layer [Section 4.5.1]:* This layer consists in the subshift X_{adR}. Recall that any configuration of this subshift exhibits a cell hierarchy. In this setting, for all n the order n cells appear periodically in the vertical and horizontal directions. Some additional marks allow the decomposition of the cells into sub-structures that we describe in this section. In each of these sub-structures specific behaviors occur. In particular, each cell is decomposed into four parts called quarters.

- *Basis layer [Section 4.5.2]:* The blue corners in the structure layer will be superimposed with random bits in $\{0,1\}$. This allows to an entropy in $[0,1/4]$ to be generated. Turing machines will control the frequency of positions where the random bits can be 1 through the use of frequency bits. This generates an entropy equal to the Π_1-computable number
$$h' = h - \frac{\lfloor 4h \rfloor}{4} \in [0, 1/4[.$$
In order to generate the entropy h, we consider $i = \lfloor 4h \rfloor$. We impose that the other positions are superimposed with random bits such that the positions where the random bit is equal to 1 have with frequency $i/3$ in this set. Hence the total frequency of the positions where the random bit can be 1 in $\mathbb{Z}^2$ is
$$\frac{i}{3}\frac{3}{4} + h' = h.$$

- *Frequency bits layer [Section 4.5.3]:*

  Each quarter is superimposed with a frequency bit. On positions with blue corners in the structure layer having frequency bit equal to 0, the random bit is equal to 0.

- *Cells coding layer [Section 4.5.4]:*

  In this layer we superimpose to the center of the cells (that we call nuclei) of each cell a symbol which specifies two adjacent quarters of the cell when $n > N$. It represents all the quarters when $n \leq N$. This symbol is called the DNA of the cell.

  The quarters represented in the DNA are called computation quarters. The other ones are called simulation quarters. In these ones the function of the machines is to simulate any degenerated behavior of the computing machines.

  Since simulation induces parasitic entropy, we choose after the construction some N such that this entropy is smaller than h. This allows programming the machines in such a way that the entropy generated by the random bits in the other quarters complements this entropy so that the total entropy is h.

- *Synchronization layer [Section 4.5.5]:* In this section, we synchronize the frequency bits of the computation quarters of all the cells having the same order.

- *Computation areas layer [Section 4.5.6]:* In this layer are specified the areas supporting the computations of the machines. The function of each position in this area amongst the following ones are also specified:

  1. transfer of information, vertical or horizontal,
  2. or the execution of one step of computation.

  This is done using signals that detect the rows and columns in a cell that do not encounter a smaller cell. The intersections of such line and column are the positions where a machine executes one step of its computation. The other positions of these lines and columns are used to transmit information.

  Since the construction of these areas can have degenerated behaviors in infinite cells, these behaviors are also simulated in simulation quarters. We use error signals in order to impose that the computation areas are well constructed in computation quarters.

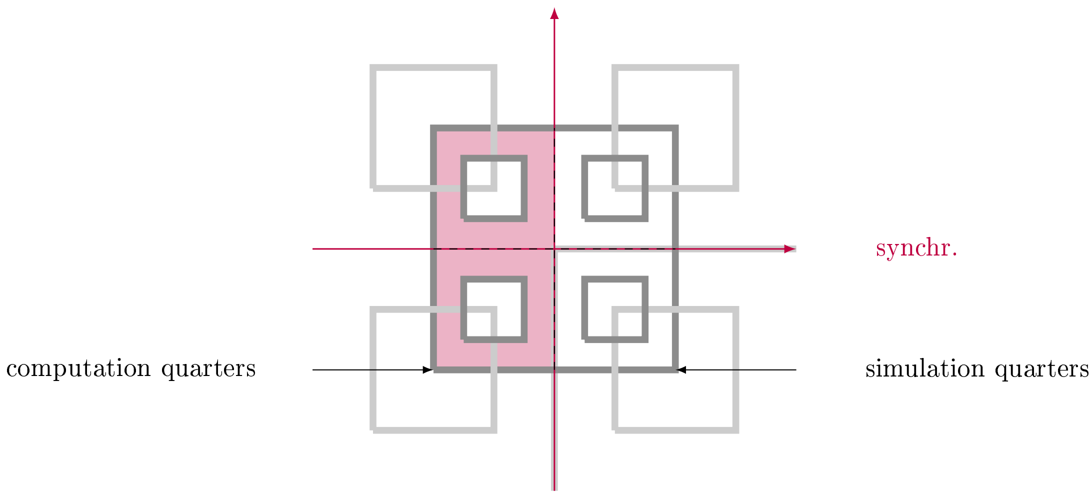


Figure 4.20: Illustration of the frequency bits identification areas (colored purple) in our construction. The squares designate some petal structures of the Robinson subshift.

- *Machines layer [Section 4.5.7]:*

  Consider $(s_k^{(n)})_{n,k} \in \{0,1\}^{\mathbb{N}^2}$ some sequence such that for all k,

$$s_k = \inf_n s_k^{(n)}.$$

  This layer supports the computations of the machines in all the quarters of each cell. Each quarter of a cell has its proper direction of time and space. The machine is programmed so that when well initialized, it writes successively for all n the bits $s_k^{(n)}$, $k = 0...n$ on the 2^kth position of its tape corresponding to a column that is just one the left of order k cells. The sequence is chosen afterwards so that the total entropy of $X_{s,N}$ is h.

  The frequency bits of order k cells corresponding to the computation quarters is transported in the column just on the right of these cells. This is done in order for the machine to have access to this information.

  If at some point one of the frequency bits is greater than a bit written by the machine in this column, then the machine enters in halting state.

  We allow the machine heads to enter in error state. However, when this happens it transmits a signal through its trajectory back in time until initialization. We forbid the coexistence of this signal with the representation of this quarter in the DNA. As a consequence, the computations of a machine are taken into account if and only if in a computation quarter where these computations have some meaning.

  The directions of time and space depend on the orientation of the quarter in the cell. In each quarter, these directions are given on Figure 4.22.

The next sections are devoted to make the proof of Theorem 1.2 more precise.

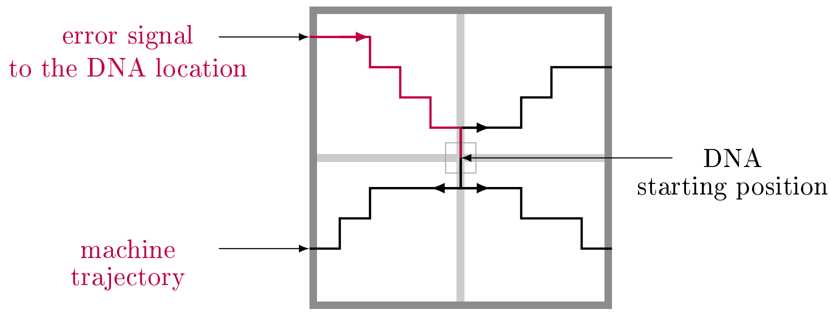


Figure 4.21: Illustration of the machines mechanisms in our construction. The gray square designates a computing unit splitted in four parts. In each of this parts evolves a computing machine departing from the center of the cell. If the machine reaches the border in error state, it triggers an error signal.

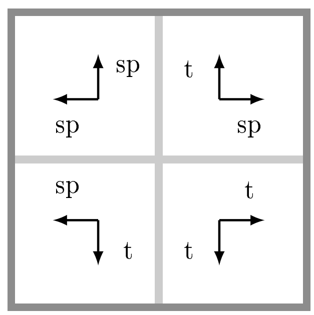


Figure 4.22: Schema of time (t) and space (sp) directions in the four different quarters of a cell.

4.5 Construction of the subshifts $X_{s,N}$

4.5.1 Structure layer

In this section, we present the structure layer and the structures observable in this layer that will be used in the following.

This layer has three sub-layers, as follows.

4.5.1.1 Specialized sub-structures of the cells

The first sublayer of the structure layer is the subshift X_{adR} presented in Chapter 3.

Here we list some structures that appear in this layer and the designation that we use for them. Let x be a configuration in this layer.

- Recall that an order n cell is the part of $\mathbb{Z}^2$ enclosed in an order $2n+1$ petal.
- The center of a cell is a red corner which is valued with 0. The position where this symbol appears is called the **nucleus** of the cell.
- In a cell the union of the column and the line containing the nucleus is called the **reticle** of the cell. In particular, the nucleus is the intersection of the reticle's arms.

- We call the set of positions in the border of a cell (defined as the position which has a neighbor outside of the cell) the **wall**.
- The set of positions in a cell that are not in the wall, in the reticle, or in another (smaller) cell included to the considered one is called the **cytoplasm**.
- We call any of the four connected parts included in the cytoplasm and delimited by the reticle and the wall a **quarter** of the cell.

Figure 4.39 gives an example of pattern that can be superimposed over an order 1 cell in the structure layer. We illustrate on this figure the structures that we listed above.

Recall that cells have the following properties:

- For all $m \geq 0$, and $i \in \{1, ..., m\}$, any order m cell contains properly 4.12^{i-1} order $m-i$ cells (meaning that these cells are not included in an order $< m$ cell). As a consequence, each order m cell contains properly 4.12^m blue corners.
- Moreover, in any configuration each order m cell (and in particular the nuclei) repeats periodically, in the horizontal and vertical directions, with period 4^{m+2}.

4.5.1.2 Coloring

In this section, we present the representation of the sub-structures of cells presented in Section 4.5.1.1. In particular, we color differently the four quarters of a cell, in order for the machines to have access to the direction of time and space in their quarter.

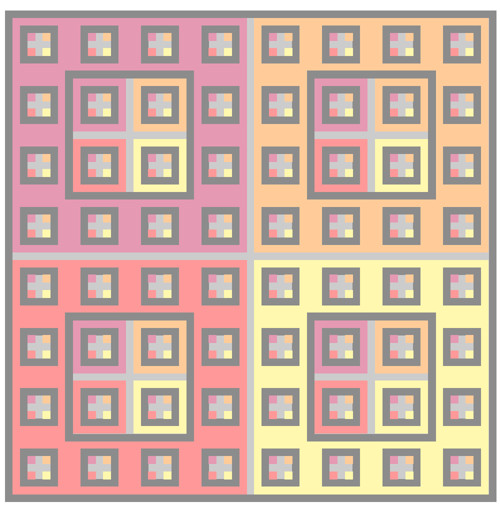
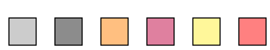


Figure 4.23: An example of pattern in the synchronization layer over an order 2 cell.

Symbols:



Each symbol corresponds to a part of the cell:

- corresponds to the reticle.
- corresponds to the walls.
- corresponds to the north east quarter.
- corresponds to the north west synchronization area.
- corresponds to the south east quarter.
- corresponds to the south west quarter.

Local rules:

1. **Localization:**
   - A position is colored with if and only if it lies in the walls of a cell.
   - the symbol can be superimposed only on red corners valued 0 or arrows symbols valued 0 for the direction of the long arrows.
   - the positions that are not amongst the type of positions specified by the two last rules are colored with else , , or .

2. **Transmission rules:**
   - Consider two vertically adjacent positions with vertical outgoing arrows (or two horizontally adjacent positions with horizontal ougoing arrows) such that the value corresponding to this direction in the two symbols is 0. Then if one of these positions is colored , then the other one is also colored .
   - Consider two adjacent positions which are not colored or . Then these two positions have the same color.

3. **Determination of the colors:**

   The idea of these rules is to determine the colors that appear in the neighborhood of particular positions in a cell. Together with the transmission rules, these rules will determine the color of any position in any configuration.

   In order to describe them, we use the vocabulary introduced in this text. However, one can translate these terms into symbols of the Robinson subshift (we do this for the second rule as an example).

   The rules are the following ones:

   - A south west (resp south east, north east, south west) corner of a cell induces the position on its north east (resp north west, south west, south east) to be colored (resp. , ,).
   - A position inside the west part of the north wall of a cell (meaning a four or six arrows symbol whose long arrows are horizontal, directed to the right, and valued 1 for the horizontal direction) (resp. east part) has its south neighbor position colored with (resp.). Its neighbor position on north is colored with the same color as the north east position and the north west position. We impose similar rules for positions in the other walls of cells.
   - A nucleus has respectively on its north west, north east, south east, south west positions the colors , , and , and this position is colored .

- A position at the center of the south wall of a cell has its north west, north and north west neighbor positions colored with , and . Similar rules are imposed for the centers of the other walls.
- On the west and east neighbor positions of a position colored , the possible pairs of colors are:
  - on the west and on the east,
  - on the west and on the east.

  Similar rules are imposed for the vertical direction.

Global behavior:

The localization rules impose directly that the walls of the finite cells are colored . With the transmission rules combined with the rules determining colors, the reticle is colored . Moreover, for any cell the positions in the same quarter are colored with else , , or . The color is determined according to the orientation of the quarter in the cell: the first (resp. second, third, fourth) color is for the south west quarter (resp. south east, north east, north west).

These rules allow the same coloration for infinite cells.

Figure 4.23 shows an example of coloration of an order 2 cell.

4.5.1.3 Synchronization net

In this section, we specify a network connecting the nuclei of order n cells for all n. This allows the synchronization of the frequency bits corresponding to computation quarters of the cells.

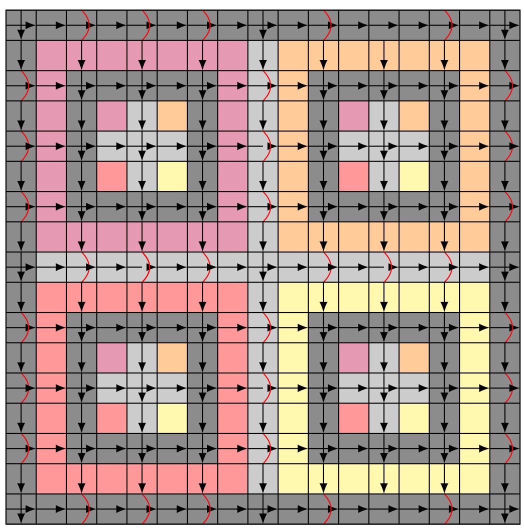
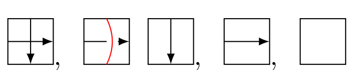


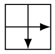
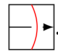
Figure 4.24: Example of a synchronization net in an order 2 cell

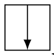
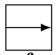
Symbols:

The symbols of this third sublayer are the following ones :



Local rules:

- The symbol  is superimposed on and only on the nuclei, corners, and centers of the walls of cells. As a consequence, when a line of arrows crosses a column of arrows on another position, then the symbol superimposed on this position is .

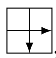
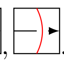
- The symbols  and  are transmitted in the direction of the arrow: north and south for the first one, east and west for the second one.

- Consider a position $\mathbf{u}$. If it is superimposed with a vertical (resp. horizontal) arrow, then the positions $\mathbf{u} \pm \mathbf{e}^1$ (resp. $\mathbf{u} \pm \mathbf{e}^2$) are not superimposed with a vertical (resp. horizontal) arrow.

Global behavior:

The first two rules build **wires** of the net, defined to be infinite columns (resp. rows) whose positions are superimposed with a vertical (resp. horizontal) arrow. These are the columns (resp. rows) intersecting corners and nuclei of cells. The other columns (resp. rows) are not superimposed with a vertical (resp. horizontal arrow): this is a consequence of the last rule.

The **intersections of the wires** are superimposed with one of the two first symbols of the alphabet:

, .

The first symbol is superimposed on nuclei and corners, where the frequency bits will be synchronized. The other intersections have the second symbol. On these positions, the frequency bits won't be synchronized. As a consequence, only cells having the same order are synchronized.

See on Figure 4.24 an example of a net superimposed on an order 1 cell.

4.5.2 Basis layer

The basis layer supports **random bits**. We recall that $i = \lfloor 4h \rfloor$.

Symbols:

The symbols of this layer are 0 and 1.

Local rules:

- if $i = 0$, then on any position $\mathbf{u}$ superimposed with a blue corner, the positions $\mathbf{u} + \mathbf{e}^1$ and $\mathbf{u} + \mathbf{e}^1 + \mathbf{e}^2$ are superimposed with the bit 0.

- if $i = 1$, the positions $\mathbf{u} + \mathbf{e}^1$ and $\mathbf{u} + \mathbf{e}^1 + \mathbf{e}^2$ are superimposed with 0, and there is no constraint on the bit on position $\mathbf{u} + \mathbf{e}^2$.

- if $i = 2$, the position $\mathbf{u} + \mathbf{e}^1$ is superimposed with 0.

- if $i = 3$, there is no constraint.

Global behavior:

In this layer are superimposed random bits in $\{0, 1\}$ on any position. According to the value of i, we impose constraints on positions that are not superimposed with a blue corner in the structure layer. We dot this in such a way that the entropy produced by these bits is the maximal possible value smaller than h.

The positions with a blue corner are the ones where the random bits will be regulated by frequency bits. These bits are themselves controlled by the machines in order to complete the entropy generated by the random bits on the other positions.

4.5.3 Frequency bits layer

Symbols:

The symbols of this layer are $0, 1$ and a blank symbol.

Local rules:

- **Localization:** The non-blank symbols are superimposed to the cytoplasm positions.
- **Synchronization on quarters:** two adjacent positions in the cytoplasm have the same frequency bit.

Global behavior:

The bits $0, 1$ are called **frequency bits**. In a quarter, these bits are equal. Hence each quarter is attached with a unique frequency bit.

4.5.4 Cells coding layer

Symbols:

and a blank symbol.

Local rules:

- The non-blank symbols are superimposed to the nuclei, the blank symbols to other positions.
- The DNA symbol  is and can only be over a nucleus of an order $\leq N$ cell.
- The others DNA symbols are over $> N$ order cells.

Global behavior:

On the nucleus of every cell is superimposed a symbol called the **DNA**. It rules the behavior of each of the machines working in the cytoplasm, telling which ones of the machines execute simulation and which ones compute.

When the order of the cell is smaller or equal to N, then all the machines compute. When the order is greater than N, two of the machines compute and the other two execute simulation.

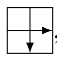
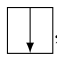
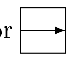
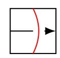
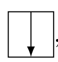
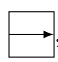
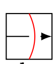
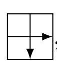
4.5.5 Synchronization layer

This layer serves for the synchronization of the frequency bits corresponding to computation quarters.

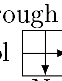
Symbols:

Elements of $\{0,1\}$ and $\{0,1\}^2$, and a blank symbol.

Local rules:

1. **Localization:**
   - The non-blank symbols are superimposed on and only on positions having a non-blank symbol in the synchronization net sublayer.
   - Positions superimposed with , , or  are superimposed in the present layer with an element of $\{0,1\}$. The positions superimposed with  are superimposed with an element of $\{0,1\}^2$.
2. **Synchronization:**
   - On a nucleus, if the DNA symbol is , then the bit in this layer is equal to the frequency bit on north east, south east, south west and north west positions.
   - When the DNA symbol is not , then the bit in this layer is equal to the frequency bits corresponding to the colors represented in the DNA symbol.
3. **Information transfer rules:**
   - Considering two adjacent positions with , or , the bits superimposed on these two positions are equal.
   - On a position with  symbol, the first bit of the ordered pair is equal to the bit of positions on north and south. The second one to the bit of positions on west and east.
   - On a position superimposed with , the bit is equal to the bit on south, west, east and north positions.

Global behavior:

Using the information contained in the nucleus (the DNA), we synchronize the two frequency bits of the computation quarters. These bits are transmitted through the wires of the synchronization net. They are synchronized on the positions having the symbol . As a consequence, when $n \leq N$, all the frequency bits of order n cells are equal. When $n > N$, the frequency bits corresponding to computation quarters are synchronized. The other two are left free and are not synchronized between cells.

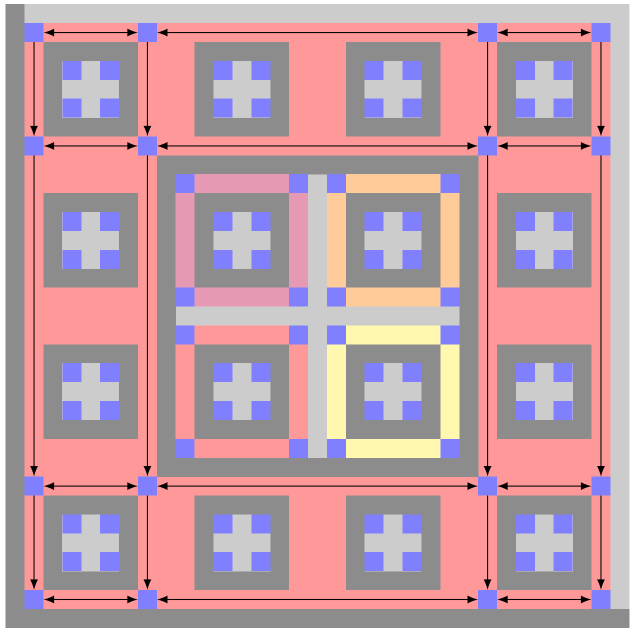


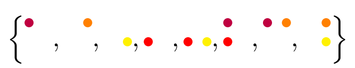
Figure 4.25: Example of a computation area in a computation quarter in an order 2 cell.

4.5.6 Computation areas layer

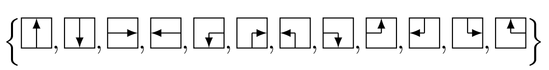
This layer specifies the function of each position of the cytoplasm relatively to the Turing machines: information transfer (vertical or horizontal) or execution of one computation step. This is done as in [Rob71]. However, the constitution of the computation areas in infinite cells is not well controlled. That is the reason why, in order to ensure the linear net gluing property, we simulate degenerated behaviors for the constitution of computation areas in the simulation quarters. We use error signals to ensure that the computation areas are well constituted in the computation quarters.

Symbols:

Elements of $\{\mathtt{in}, \mathtt{out}\}^2 \times \{\mathtt{in}, \mathtt{out}\}^2$, elements of



elements of



and a blank symbol.

The first set corresponds to signals that propagate horizontally and vertically in the cytoplasm. The second set corresponds to error signals propagating on the reticle and to the nucleus. The last ones correspond to the propagation of the error signals through the walls.

Local rules:

- **Localization:**

– The elements of $\{\mathtt{in}, \mathtt{out}\}^2 \times \{\mathtt{in}, \mathtt{out}\}^2$ are superimposed on and only on positions in the cytoplasm. The first two coordinates are associated to the horizontal direction, and the second two to the vertical direction.

– The elements of  can be superimposed only on reticle positions.

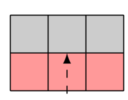
– All the other positions are superimposed with the blank symbol.

- **Transmission of the cytoplasm signals:** on a cytoplasm position $\mathbf{u}$, the two first coordinates of the symbol are transmitted to the positions $\mathbf{u} \pm \mathbf{e}^1$ and the second two are transmitted to the positions $\mathbf{u} \pm \mathbf{e}^2$, when these positions are in the cytoplasm.

  In a computation quarter, these are signals allowing the lines and columns of the cells which do not intersect a smaller cell to be specified. In the first (resp. second) ordered pair, the symbol $\mathtt{in}$ for the first coordinate corresponds to the fact that the position is in a segment of row (resp. column) originating from the inside of the cell wall. The symbol $\mathtt{out}$ corresponds to the fact that the position is in a segment of row (resp. column) originating from the outside of the cell wall. For the second coordinate, these symbols have the same signification concerning the end of the segment instead. The next rules impose that when near the pertinent parts of a cell and inside it, if a symbol $\mathtt{in} - \mathtt{out}$ does not correspond to the nature of the origin or end at this position, this triggers an *error signal*. When outside the cell, the origin or end is imposed - thus not triggering an error signal. These rules are presented for positions in the **red quarter**: similar rules are imposed for the other ones.

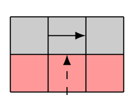
- **Triggering error signals (inside the wall and reticle):** On a position $\mathbf{u}$ in the horizontal arm of the reticle (specified by having a reticle symbol different from the nucleus and having a reticle position on the right and on the left), if the position $\mathbf{u} - \mathbf{e}^2$ has its second ordered pair having second coordinate equal to $\mathtt{out}$, then the red quarter is represented in the symbol superimposed on position $\mathbf{u}$.

  For instance, the pattern

  $(*, \mathtt{out})$,

  where $*$ means any symbol, and the ordered pair represented in $\{\mathtt{in}, \mathtt{out}\}^2$ is the second one, implies the following:

  $(*, \mathtt{out})$.

  On a position $\mathbf{u}$ in the vertical arm of the reticle (specified by having a reticle symbol different from the nucleus and having a reticle position on the top and bottom), if the position $\mathbf{u} - \mathbf{e}^1$ has its first ordered pair having second coordinate equal to $\mathtt{out}$, the red quarter is represented in the symbol superimposed on position $\mathbf{u}$.

- On a position $\mathbf{u}$ in the horizontal part of the wall (specified by having a wall symbol different from the corner, and having a wall position on left and right), if the position $\mathbf{u} + \mathbf{e}^2$ has its second ordered pair having first coordinate equal to `out`, then the symbol on the position $\mathbf{u}$ is an arrow symbol

 or .

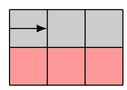
On a position $\mathbf{u}$ in the vertical part of the wall (specified by having a wall symbol different from the corner, and having a wall position on top and bottom), if the position $\mathbf{u}+\mathbf{e}^1$ has its second ordered pair having first coordinate equal to `out`, then the symbol on the position $\mathbf{u}$ is an arrow symbol

 or .

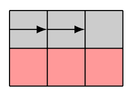
- **Enforcing cytoplasm signals (outside the wall):**

  Considering a wall position $\mathbf{u}$ which is on the west (resp. east, north, south) wall of a cell, the position $\mathbf{u}-\mathbf{e}^1$ (resp. $\mathbf{u}+\mathbf{e}^1$, $\mathbf{u}+\mathbf{e}^2$, $\mathbf{u}-\mathbf{e}^2$ has the second coordinate of its second ordered pair (resp. first coordinate and second ordered pair, first coordinate first ordered pair, second coordinate first ordered pair) equal to `out`.

- **Propagation of error signals**. An arrow symbol propagates in the direction pointed by the arrow on the wall, while the next position in this direction is not near a reticle position, as in the following pattern:

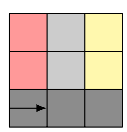
  

  implies the following one:

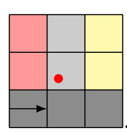
  

- On a position $\mathbf{u}$ of the north (resp. east, south, west) arm of the reticle, specified by the colors on the sides, if the position $\mathbf{u}-\mathbf{e}^1$ (resp. $\mathbf{u}-\mathbf{e}^1$, $\mathbf{u}+\mathbf{e}^2$, $\mathbf{u}+\mathbf{e}^1$) is not the nucleus, then the symbol on this position contains the symbol on position $\mathbf{u}$.

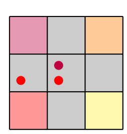
- **Connection between error signals:** when on a position $\mathbf{u}$ on the wall which is near a position on the reticle, if one of the wall symbols aside contains an error symbol, then the reticle position has an error symbol where the corresponding quarter is represented. For instance, the pattern

  implies the following:

  .

- **Forbidding wrong error signals**. On any of the four reticle positions around the nucleus, there can not be a symbol that contains a color which is in the DNA. For instance, the following pattern is forbidden:

Global behavior:

In any quarter of a cell, the segment of rows and columns are colored with an ordered pair of symbols in $\{\texttt{in}, \texttt{out}\}$. One for the origin of the segment, the other one for the end of it. If it originates from or ends at the outside of a cell, then the corresponding symbol is forced to be `out`.

Moreover, when near the walls or the reticle and inside the corresponding cell, if the corresponding symbol is `out` then an error signal is triggered and propagates to the nucleus. On the walls, a propagation direction is chosen. In the reticle, the error signals contain the information about the quarter where the error was detected. Around the nucleus, we forbid an error signal to come from a computation quarter.

In a *computation quarter* of a cell, each row which does not intersect a smaller cell has first ordered pair equal to $(\texttt{in}, \texttt{in})$, since it originates inside the cell, and ends inside, on the reticle. Each column which does not intersect a smaller cell has second ordered pair equal to $(\texttt{in}, \texttt{in})$. The ordered pairs on other segments of rows or columns are determined in a similar way, according to their origin and end. This is enforced by the propagation of error signals to the nucleus.

The positions marked with $((\texttt{in}, \texttt{in}), (\texttt{in}, \texttt{in}))$ are called **computation positions**. The ones that have first (resp. second) ordered pair equal to $(\texttt{in}, \texttt{in})$ and second (resp. first) not equal to $(\texttt{in}, \texttt{in})$ are **horizontal transfer positions** (resp. vertical transfer positions). See Figure 4.25 for a representation of a computation area in the red quarter of an order two cell. On this figure, computation positions are represented by a blue square. Vertical and horizontal transfer positions by arrows in this direction.

Remark 29. *These mechanisms can not be easily simplified, since an infinite row or a column can not "know" if it is a free row or column of its infinite cell. Moreover, from the division of the cells it is difficult to code this with a hierarchical process.*

Remark 30. *In the literature, most of the constructions using substitutions include the construction of the computation areas [HM10] with substitution rules. However, in order to get the net gluing property, and furthermore the block gluing property, we need a more flexible construction of the computation areas. The method presented above was used initially in the construction of Robinson for his undecidability result [Rob71].*

4.5.7 The machines (RNA)

In this section, we present the implementation of Turing machines that will check that the frequency bit of level n cells are equal to s_n, for all n.

In order to have the linear block gluing property, we have to adapt the Turing machine model in order to simulate each possible degenerated behavior of the machines. This is done as follows: in each of the quarters of a cell, a machine is implemented. For this machine, the directions of space and time are as on Figure 4.26: the rules of the machine will depend on the color of the quarter. Moreover, for each of the quarters, we initialize the tape with elements of $\mathcal{A} \times \mathcal{Q}$. The set $\mathcal{Q}$ is the state set of the machine and $\mathcal{A}$ its alphabet. Machine heads can enter on the two sides of the computation area. Signals will be used to verify that in the computation quarters the machine is well initialized. This means that no head enters on the sides, and on the initial row there is a unique machine head on the position near the nucleus in initial state. Moreover, all the letters in $\mathcal{A}$ are blank.

As usually in this type of constructions, the tape is not connected. Between two computation positions, the information is transported. In our model, each computation position takes as input up to four symbols coming from bottom and the sides. It outputs up to two symbols to the top and sides. Moreover, we add special states to the definition of Turing machine. We do this in order to manage

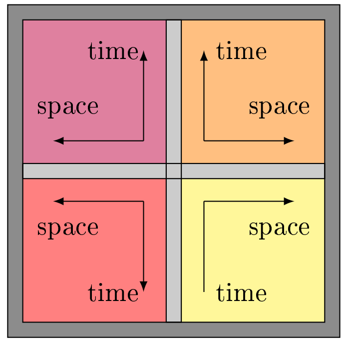


Figure 4.26: Schema of time and space direction in the four different quarters of a cell.

the presence of multiple machine heads. We describe this model in Section 4.5.7.1, and then show how to implement it with local rules in Section 4.5.7.2.

If a machine head enters an error state, this triggers an error signal that propagates through the trajectory of the machine. This signal is taken into account only for computation quarters.

4.5.7.1 Adaptation of computing machines model to linear block gluing property

In this section we present the way computing machines work in our construction. The model that we use is adapted in order to have the linear block gluing property, and is defined as follows:

Definition 4.29. *A* ***computing machine*** $\mathcal{M}$ *is some tuple* $= (\mathcal{Q}, \mathcal{A}, \delta, q_0, q_e, q_s, \#)$. *The set* $\mathcal{Q}$ *is the state set,* $\mathcal{A}$ *the alphabet,* q_0 *the initial state, and* $\#$ *is the blank symbol, and*

$$\delta : \mathcal{A} \times \mathcal{Q} \to \mathcal{A} \times \mathcal{Q} \times \{\leftarrow, \to, \uparrow\}.$$

The other elements q_e, q_s *are states in* $\mathcal{Q}$. *They are such that for all* $q \in \{q_e, q_s\}$ *and for all* a *in* $\mathcal{A}$, $\delta(a, q) = (a, q, \uparrow)$.

The special states q_e, q_s in this definition have the following meaning:

- error state q_e: a machine head enters this state when it detects an error or when it collides with another machine head.

  This state is not forbidden in the subshift, but this is replaced by the sending of an error signal. We forbid the coexistence of the error signal with a well initialized tape. The machine stops moving when it enters this state.

- shadow state q_s: this state corresponds to the absence of head. We need to introduce this state so that the number of possible space-time diagrams in finite cells has a closed form.

Any Turing machine can be transformed in such a machine by adding some state q_s verifying the properties listed above.

When the machine is well initialized, none of these states and letters will be reached. Hence this machine behave as the initial one. As a consequence, one can consider that the machine we used has these properties.

In this section, we use a machine which successively for all $n \geq 0$ writes the bits $s_k^{(n)}$, $k = 1...n$, on positions $p_k = 2^k$ (which is a computable function). This position corresponds to the number of the first active column from left to right which is just on the right of an order n two dimensional cell on a face amongst active columns

Recall that s is the Π_1-computable sequence defined at the beginning of the construction. The sequence $(s_k^{(n)})$ is a computable sequence such that for all k, $s_n = \inf_k s_k^{(n)}$.

4.5.7.2 Implementation of the machines

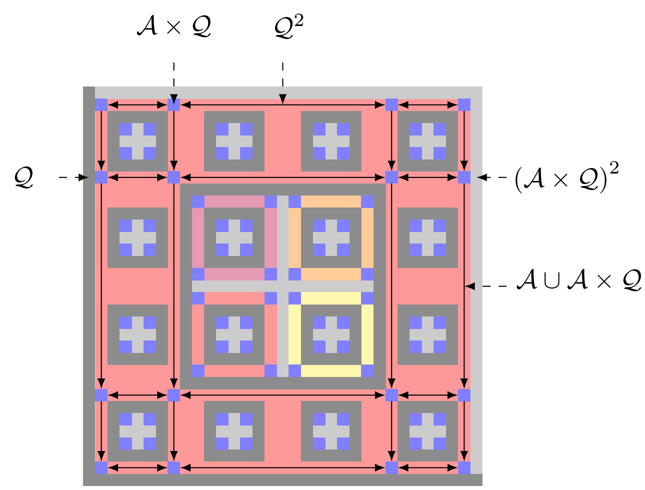


Figure 4.27: Localization of the machine symbols in the red quarter of an order two cell.

In this section, we describe the second sublayer of this layer.

Symbols:

The symbols are elements of the sets $\mathcal{A} \times \mathcal{Q}$, $\mathcal{A}$, $\mathcal{Q}^2$, $\mathcal{Q}$, $(\mathcal{A} \times \mathcal{Q})^2$, and a blank symbol.

Local rules:

- **Localization:** the non-blank symbols are superimposed on information transfer rows and columns, as well as positions corresponding to information transfer rows and columns on the arms of the reticle and the east and west walls. More precisely:
  - the possible symbols for information transfer columns are elements of the sets $\mathcal{A}$ and $\mathcal{A} \times \mathcal{Q}$. The elements of $\mathcal{A} \times \mathcal{Q}$ are on computation positions. The other ones on the other positions of these lines and columns.
  - the positions on the vertical (resp. horizontal) arms of the reticle corresponding to an information transfer line are colored with an element of $\mathcal{Q}^2$ (resp. $(\mathcal{A} \times \mathcal{Q})^2$). The first coordinate corresponds to the machine heads entering in the west quarter. The second one corresponds to machine heads entering in the east one (resp. machine head and letter entering in the north and south ones).
  - on the west and east walls, the symbols are in $\mathcal{Q}$. They correspond to machine heads entering in the adjacent quarter. See an illustration on Figure 4.27.
- **Transmission:**

  Along the rows and columns, the symbol is transmitted while not on computation positions.

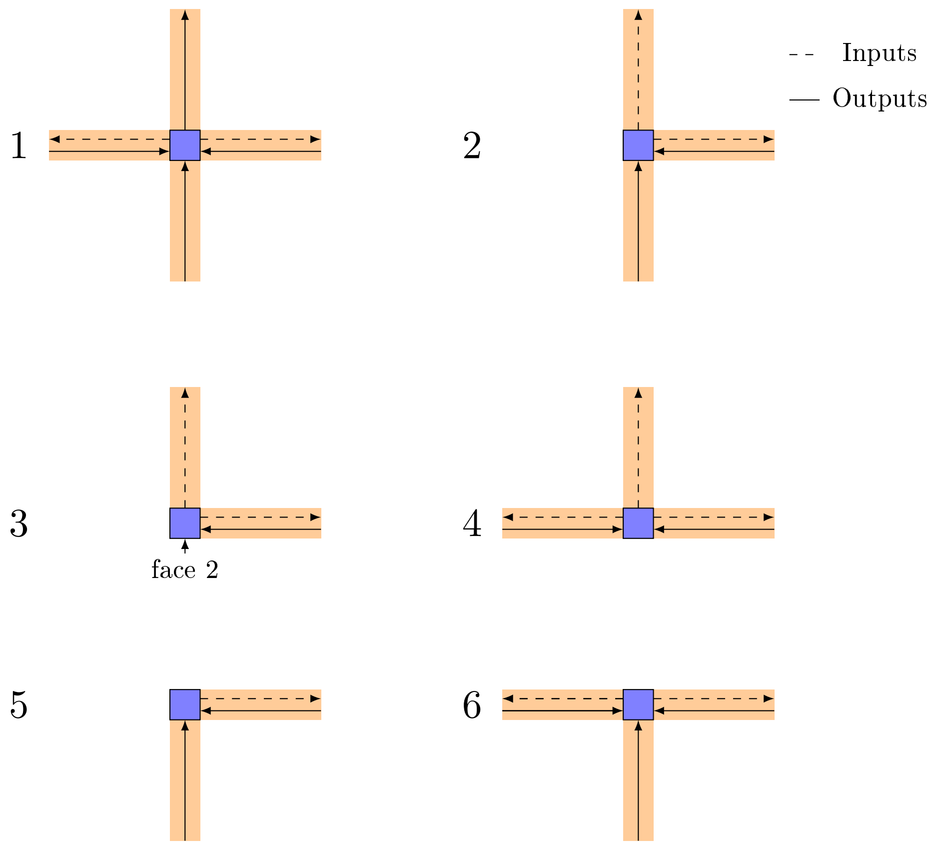


Figure 4.28: Schema of the inputs and outputs directions when inside the area (1) and on the border of the area (2,3,4,5,6).

- **Computation positions rules:**

  Consider some computation position. These rules depend on the orientation of the quarter in the cell. We describe them in the north east quarter. The rules in the other quarters are obtained by symmetry, respecting the orientation of time and space given on Figure 4.26.

  For such a position, the **inputs** include:

  1. the symbols written on the south position,
  2. the first symbol written on the west position (except in the leftmost column, where the input is the second symbol of the west position),
  3. and the second symbol on the east position (except when in the rightmost column, where the input is the unique symbol written on east position).

  The **outputs** include:

  1. the symbols written on the north position when not in the topmost row,
  2. the second symbol of the west position (when not in the leftmost column),
  3. and the first symbol on the east position (when not on the rightmost column).

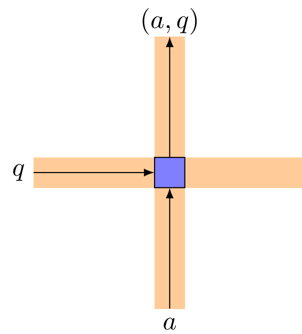


Figure 4.29: Illustration of the standard rules (1).

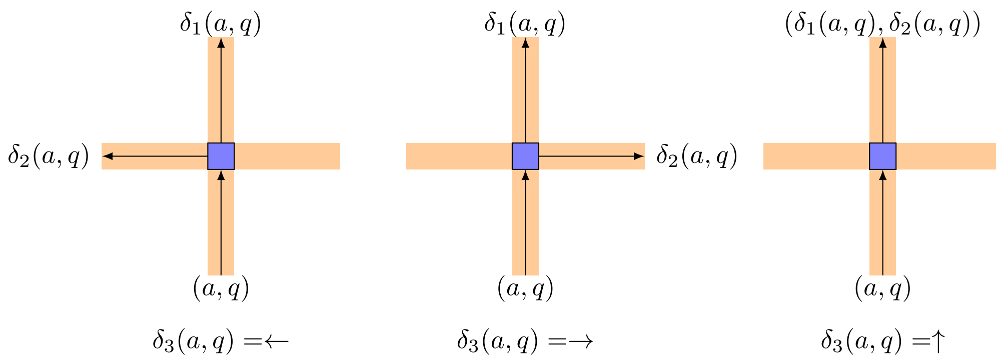


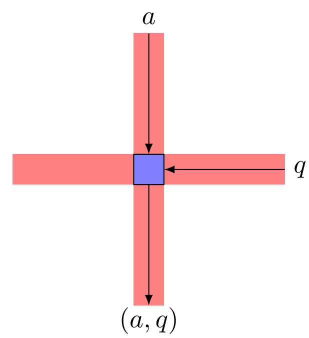
Figure 4.30: Illustration of the standard rules (2).

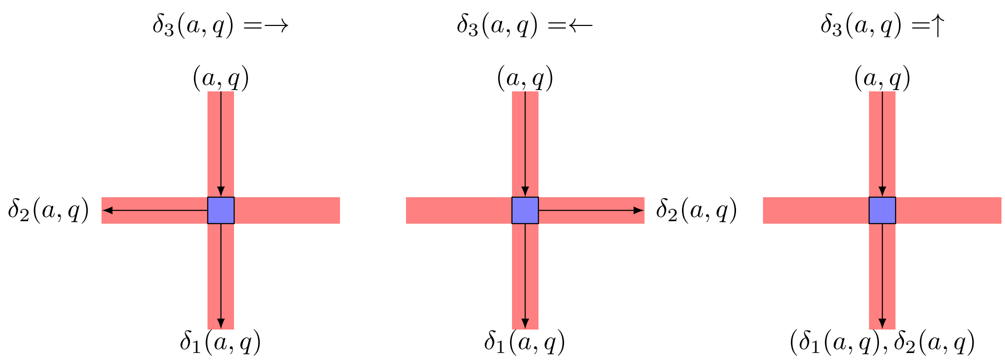
Moreover, on the row near the reticle, the inputs from inside the area are always the shadow state q_s. The input from the bottom is free. As a consequence the ordered pair written on the the position is also free. This is also true for the elements of $\mathcal{Q}$ on the computation positions in the leftmost and rightmost columns and the triple of symbols written on the position near the nucleus.

See Figure 4.28 for an illustration.

On the first row, all the inputs are determined by the counter and by the above rule. Then each row is determined from the adjacent one on the bottom and the inputs on the sides. This is due to the following rules, which on each computation position determine the outputs from the inputs:

1. **Collision between machine heads:** if there are at least two elements of $\mathcal{Q} \backslash \{q_s\}$ in the inputs, then the computation position is superimposed with (a, q_e). The output on the top (when this exists) is (a, q_e), where a is the letter input below. The outputs on the sides are q_s. When there is a unique symbol in $\mathcal{Q} \backslash \{q_s\}$ in the inputs, this symbol is called the machine head state (the symbol q_s is not considered as representing a machine head).
2. **Standard rule:**
   (a) when the head input comes from a side, then the functional position is superimposed with (a, q). It outputs the ordered pair (a, q) above, where a is the letter input under,





and q the head input. The other outputs are q_s. See Figure 4.29 for an illustration of this rule.

(b) when the head input comes from under, the output is $\delta_1(a,q)$ above when the $\delta_3(a,q)$ is in $\{\rightarrow, \leftarrow\}$ and $(\delta_1(a,q), \delta_2(a,q))$ when $\delta_3(a,q) = \uparrow$. The head output is in the direction of $\delta_3(a,q)$ when this output direction exists, and equal to $\delta_2(a,q)$ when this direction is in $\{\rightarrow, \leftarrow\}$. The other output is q_s. See Figure 4.30 for an illustration.

The computation positions rules in the other quarters are similar. These rules in a purple quarter are obtained by reversing west and east, in the red one by reversing west and east and moreover north and south, and in the yellow one by reversing north and south. For instance the previous schemata are changed to the following one in the red quarter.
This corresponds to changing the direction of time and space of evolution of the Turing machine, as abstracted on Figure 4.26.

3. **Collision with border:** When the output direction does not exist, the output is (a, q_e) on the top. The output on the side is q_s. The computation position is superimposed with (a,q).
4. **No machine head:** when all the inputs in $\mathcal{Q}$ are q_s, then the output above is in $\mathcal{A}$ and equal to a.

Global behavior:

In each of the quarters of any cell is implemented a computing machine according to our model, with multiple machine heads on the initial tape and entering in each row. In the next section, we will impose that in the computation quarters there are no machine heads entering on the sides. We also impose that the tape is well initialized. This is done using signals. As a consequence, in these quarters the computations are as intended. This means that a Turing machine writes successively the bits $s_k^{(n)}$ on the $p_k = 2^k$th column of its tape (in order to impose the value of the frequency bits). It enters in the error state q_e when it detects an error - meaning that the corresponding frequency bit in the column just on the right is greater than the written bit.

When this is not the case, the computations are determined by the rules giving the outputs on computation positions from the inputs. When there is a collision of a machine head with the border it enters into state q_e and when heads collide, they fusion into a unique head in state q_e.

4.5.7.3 Empty tape and sides signals

This sublayer serves for the propagation of a signal which detects if the initial tape of a machine is well initialized, and if a machine head enters on a side.

Symbols:

Elements of {🟥,🟩}^{2}, elements of {🟥,🟩} and a blank symbol.

Local rules:

- **Localization:** the non-blank symbols are superimposed on and only on the arms of the reticle, and the west and east walls. The east and west walls are colored with elements of {🟥,🟩}, and the reticle with element of {🟥,🟩}^{2}.
- **Triggering the signal:** the topmost and bottommost positions of the two walls are superimposed with 🟩.
- **Transmission rules:**
  1. **In the walls:**
     - On the north (resp. south) part of one of the walls, the symbol 🟩 propagates upwards while in the wall. It propagates downwards (resp. upwards) while not encountering a symbol in $\mathcal{Q}\backslash\{q_s\}$. When this is is the case, the color becomes 🟥.
     - The symbol 🟥 propagates downwards (resp. upwards) in the north (resp. south) part of the wall, and upwards (resp. downwards) while not encountering a symbol in $\mathcal{Q}\backslash\{q_s\}$.
     - The center of the wall is colored with an ordered pair of colors. The first one is equal to the color of the north position. The second one is equal to the color of the south position.

     With words, a signal propagates through the north (resp. south) part of the wall. This signal is triggered in state 🟩. When it detects the first symbol in $\mathcal{Q}\backslash\{q_s\}$, it changes its state which becomes 🟥. This information is transmitted to the center of the wall.
  2. **In the reticle:** The rules for the reticle are similar, except that:
     - there are two signals for each arm, one for each adjacent quarter.
     - the propagation direction is to the east for the west arm, and to the west for the east arm. The case of vertical arms is similar as the case of walls.
     - when the signal starts in state 🟩: on the west arm (resp. east one) each signal detects the first symbol from left to right (resp right to left) different from $(\#, q_s)$ when not on the rightmost position (resp. leftmost one), and different from $(\#, q_0)$ when on this position. These symbols correspond to the quarter associated with this signal.

  - when it starts in state , the arm just transmits this information to the nucleus.
  - In the vertical arms, the signals detect the first symbol different from q_s, from south to north for the south arm, and from north to south for the north arm.

- **Computation quarters are initialized with empty tape and sides:** considering a nucleus $\mathbf{u}$, the symbol on position $\mathbf{u} + \mathbf{e}^1$ has first (resp. second) coordinate equal to if the orange (resp. yellow) quarter is represented in the DNA.
- the symbol on position $\mathbf{u} - \mathbf{e}^1$ has first (resp. second) coordinate equal to if the purple (resp. red) quarter is represented in the DNA.
- There are similar conditions for the symbols on positions $\mathbf{u} \pm \mathbf{e}^2$.

Global behavior:

These rules induce the propagation of a signal triggered in state which detects, for each of the quarter, if the sides and the tape are well initialized: if this is not the case, then the signal detects an error. It sends this information, which corresponds to state , to the nucleus through the reticle. We forbid this signal to come from a quarter which is represented in the DNA. As a consequence, the computation quarters are well initialized. The simulation quarters are left free.

4.5.7.4 Error signals

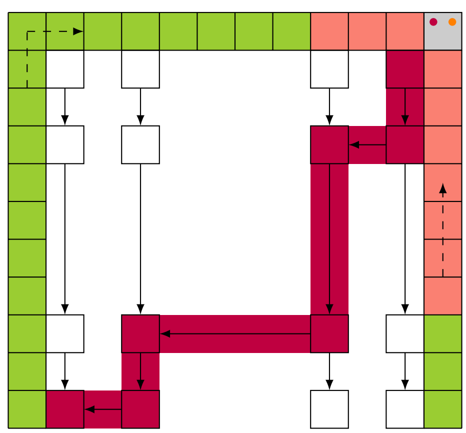


Figure 4.31: Schema of a possible trajectory of a machine in a simulation quarter. The empty tape and sides signals represented correspond to the south west quarter. The dashed arrows represent the propagation direction of these signals.

Symbols:

This sub-layer has the following symbols: and .

Local rules:

- **Localization:** the symbol can be superimposed only on positions having in its machine symbol a part in $\mathcal{Q} \backslash \{q_s\}$.

- **Transmission:** for two adjacent vertical transfer or horizontal transfer positions, the symbols in this sublayer are the same.
- when on a computation position $\mathbf{u}$, if two of the positions $\mathbf{u} \pm \mathbf{e}^1$ and $\mathbf{u} \pm \mathbf{e}^2$ have a part in $\mathcal{Q}\backslash\{q_s\}$, these two positions have the same symbol in this layer.
- **Triggering the error signal when on halting state:** a position with a symbol having a part equal to q_h is superimposed with $\blacksquare$.
- **Machine heads can not enter in error state:** if $\mathbf{u}$ is a nucleus position and the red quarter (resp. yellow, orange, purple ones) is represented in the DNA, then the position $\mathbf{u} - \mathbf{e}^2 - \mathbf{e}^1$ (resp. $\mathbf{u} - \mathbf{e}^2 + \mathbf{e}^1$, $\mathbf{u} + \mathbf{e}^2 + \mathbf{e}^1$, $\mathbf{u} + \mathbf{e}^2 - \mathbf{e}^1$) can not be superimposed with $\blacksquare$

Global behavior:

When a machine enters in error state, then it sends through its trajectory an error signal (represented by the symbol $\blacksquare$). See Figure 4.31 for a schematic example of a possible trajectory of a machine. In a computation quarter, since the tape is well initialized and the error signal is forbidden, this means that the machines in such a quarter effectively forbid the frequency bits f_n to be different from s_n.

4.6 Entropy formula for the entropy of $X_{s,N}$ and choices of the parameters.

In this section, we prove a formula for the entropy of the subshifts $X_{s,N}$. Using this formula, we describe how to choose N so that the entropy generated by simulation is small enough. Then s is chosen in order to complete this entropy so that the entropy of $X_{s,N}$ is equal to h.

The formula relies on the density of the observable structures in the subshift X_{adR}. We compute these densities in order to stress how the formula relies on them.

4.6.1 Density properties of the subshift X_{adR}

In this section we define the density of a subset of $\mathbb{Z}^2$ and compute the density of some subsets related to the subshift X_{adR}.

Definition 4.30. *Let Λ be a subset of $\mathbb{Z}^2$. Denote for $n \geq 1$:*

$$\mu_n(\Lambda) = \left|\Lambda \cap [\![-n, n]\!]^2\right|.$$

The upper and lower densities of Λ in $\mathbb{Z}^2$ are defined as respectively

$$\overline{\mu}(\Lambda) = \limsup_n \frac{\mu_n(\Lambda)}{(2n+1)^2}$$

and

$$\underline{\mu}(\Lambda) = \liminf_n \frac{\mu_n(\Lambda)}{(2n+1)^2}.$$

When the limit exists, it is called the density of Λ, and denoted $\mu(\Lambda)$.

Lemma 4.31. *1. Let Λ be a subset of $\mathbb{Z}^2$ having a density. Then $\Lambda^c = \mathbb{Z}^2\backslash\Lambda$ has a density and*

$$\mu(\Lambda^c) = 1 - \mu(\Lambda).$$

2. *Let $(\Lambda_k)_{k=1..m}$ be a finite sequence of subsets of $\mathbb{Z}^2$ such that for all $j \neq k$, $\Lambda_j \cap \Lambda_k = \emptyset$. Then the set $\bigcup_{k=1}^m \Lambda_k$ has density equal to*

$$\mu\left(\bigcup_{k=1}^m \Lambda_k\right) = \sum_{k=1}^m \mu(\Lambda_k).$$

Proof. 1. For all n, the number of elements of Λ^c in $[\![-n,n]\!]^2$ is $(2n+1)^2 - |(\Lambda \cap [\![-n,n]\!]^2|$. Hence we have that

$$\mu_n(\Lambda^c) = 1 - \mu_n(\Lambda).$$

This means that Λ^c has a density equal to $1 - \mu(\Lambda)$.

2. For all n, the number of elements of $\bigcup_{k=1}^m \Lambda_k$ in $[\![-n,n]\!]^2$ is the sum of the numbers of elements in Λ_k, $k = 1...m$. An immediate consequence is that

$$\mu_n\left(\bigcup_{k=1}^m \Lambda_k\right) = \sum_{k=1}^m \mu_n(\Lambda_k).$$

Hence the density of the set $\bigcup_{k=1}^m \Lambda_k$ is indeed the sum of the densities of the sets Λ_k, $k = 1..m$. □

Let x be some configuration in the subshift X_{adR}. For all $k \geq 0$, $\Lambda_k(x)$ denotes the set of positions that are included in an order k cell, not included in any smaller cell, and on which a blue corner is superimposed. Moreover, denote the set of positions on which is not superimposed a blue corner $\Lambda_*(x)$.

Lemma 4.32. *For all x in the subshift X_{adR}, we have the following:*

1. *the density $\mu(\Lambda_k(x))$ exists for all $k \geq 0$ and*

$$\mu(\Lambda_k(x)) = \frac{3^k}{4^{k+2}}$$

and the convergence of the functions $x \mapsto \mu_n(\Lambda_k(x))$ is uniform.

2. *the set $\Lambda_*(x)$ has a density equal to*

$$\mu(\Lambda_*(x)) = \frac{3}{4}$$

and the convergence of the functions $x \mapsto \mu_n(\Lambda_*(x))$ is uniform.

3. *for all $m \geq 0$, the set $\left(\bigcup_{k=0}^m \Lambda_k(x)\right)^c$ has a density equal to*

$$\mu\left(\left(\bigcup_{k=0}^m \Lambda_k(x)\right)^c\right) = \frac{1}{4}\frac{3^{m+1}}{4^{m+1}}.$$

Proof. 1. From the form of the subshift X_{adR}, for any configuration x and for all n the set $[\![-n,n]\!]^2$ can be covered with a number smaller than $(\lceil \frac{2n+1}{2.4^{k+1}} \rceil + 1)^2$ of 2.4^{k+1}-blocks centered on an order k cell in the configuration x. In each of these blocks there are exactly 4.12^k positions in $\Lambda_k(x)$ (see the properties listed in Section 4.5.1.1). Moreover, such a pattern contains a number of translates of $[\![0, 2.4^{k+1}]\!]^2$ which is at least $(\lfloor \frac{2n+1}{2.4^{k+1}+1} \rfloor - 1)^2$. As a consequence, for all n, k,

$$4.12^k . \frac{(\lfloor \frac{2n+1}{2.4^{k+1}} \rfloor - 1)^2}{(2n+1)^2} \leq \frac{\mu_n(\Lambda_k)}{(2n+1)^2} \leq 4.12^k \frac{(\lceil \frac{2n+1}{2.4^{k+1}} \rceil + 1)^2}{(2n+1)^2}.$$

This implies that

$$\frac{\mu_n(\Lambda_k(x))}{(2n+1)^2} \to \frac{12^k}{16^{k+1}} = \frac{3^k}{4^{k+2}}.$$

2. Moreover, the set $[\![-n,n]\!]^2$ is covered by at most $(\lceil \frac{2n+1}{2} \rceil + 1)^2$ blocks on $[\![0,1]\!]^2$ such that the symbol on position $(0,0)$ is a blue corner. This set contains at least a number $(\lfloor \frac{2n+1}{2} \rfloor - 1)^2$ of translates of $[\![0,1]\!]^2$. In each of these squares the number of positions in $\Lambda_*(x)$ is 3. This implies that
$$3\frac{(\lfloor \frac{2n+1}{2} \rfloor - 1)^2}{(2n+1)^2} \leq \frac{\mu_n(\Lambda_*(x))}{n} \leq 3\frac{(\lfloor \frac{2n+1}{2} \rfloor - 1)^2}{(2n+1)^2},$$
and we deduce that
$$\mu(\Lambda_*(x)) = \frac{3}{4}.$$

3. From the second point of Lemma 4.31,
$$\mu\left(\bigcup_{k=0}^{m} \Lambda_k(x)\right) + \mu\left(\left(\bigcup_{k=0}^{m} \Lambda_k(x)\right)^c\right) + \mu\left(\Lambda_*(x)\right) = 1.$$
As a consequence, using the two first points in the statement of the lemma and the second point of Lemma 4.31,
$$\mu\left(\left(\bigcup_{k=0}^{m} \Lambda_k(x)\right)^c\right) = \frac{1}{4} - \sum_{k=0}^{m} \frac{3^k}{4^{k+2}} = \sum_{k=0}^{+\infty} \frac{3^k}{4^{k+2}} - \sum_{k=0}^{m} \frac{3^k}{4^{k+2}} = \sum_{k=m+1}^{+\infty} \frac{3^k}{4^{k+2}} = \frac{3^{m+1}}{4^{m+2}}.$$

□

4.6.2 A formula for the entropy depending on the parameters

In this section we prove a formula for the entropy depending on s and N.

Definition 4.33. *Let $(a_n)_n$ be a sequence of non-negative numbers. The series $\sum a_n$ converges at a computable rate when there is a computable function $n : \mathbb{N} \to \mathbb{N}$ (the rate) such that for all $t \in \mathbb{N}$,*
$$\left| \sum_{n \geq n(t)} a_n \right| \leq 2^{-t}$$

Remark 31. *Let $(a_n)_n$ and $(b_n)_n$ be two sequence of non-negative numbers such that for all n, $a_n \leq b_n$. If the series $\sum b_n$ converges at a computable rate, then the series $\sum a_n$ also converges at computable rate.*

Remark 32. *Let $(a_n)_n$ be some sequence of real numbers. If the series $\sum a_n$ converges at computable rate, then $\sum_{n=0}^{+\infty} a_n$ is a computable number.*

Lemma 4.34. *There exists a sequence $(\kappa_k)_{k\geq 0}$ of non-negative real numbers such that the series $\sum_k \kappa_k$ converges at computable rate, and the entropy of the subshift $X_{s,N}$ is*
$$\boxed{h(X_{s,N}) = \frac{\lfloor 4h \rfloor}{4} + \sum_{k=0}^{N} \frac{3^k}{4^{k+2}} s_k + \frac{1}{2} \sum_{k=N+1}^{+\infty} \frac{3^k}{4^{k+2}} s_k + \sum_{k=N+1}^{+\infty} \kappa_k}.$$

Proof. Let us prove that

$$
\begin{aligned}
h(X_{s,N}) \quad = \quad & \sum_{k=N+1}^{+\infty} \left(\frac{1}{32.16^k} + \frac{1}{32.8^k} \log_2(|\mathcal{A}|) + \frac{3}{32.8^k} \log_2(|\mathcal{Q}|) + \frac{1}{8.4^k} + \frac{\log_2(\kappa_k^*)}{4.16^{k+1}} \right) \\
& + \frac{1}{2} \sum_{k=N+1}^{+\infty} \frac{3^k}{4^{k+2}} (1 + s_k) \\
& + \sum_{k=0}^{N} \frac{3^k}{4^{k+2}} s_k \\
& + \frac{\lfloor 4h \rfloor}{4}
\end{aligned},
$$

where $(\kappa_k^*)_k$ is a computable sequence of integers.

Number of pattern on the proper blue corners positions in a cell:

For all $k > N$, the number of globally admissible patterns on the set of proper positions of an order k cell is equal to

$$4.|\mathcal{A}|^{2.2^k} |\mathcal{Q}|^{6.2^k}.(2^{12^k} + 1)^2.2^{2.12^k s_k}.2^{2.4.(4^k - 1)}.\kappa'(k).$$

Indeed:

1. The factor 4 corresponds to the number of possibilities for the DNA symbol on the nucleus of this cell [*See Section 4.5.4*].

2. The factor $|\mathcal{A}|^{2.2^k} |\mathcal{Q}|^{6.2^k}$ corresponds to the number of possibilities for filling the initial tapes of the two simulation quarters and the set of states of the machine heads entering on the two sides. Let us recall that the number of columns (resp. lines) in of quarter, that do not intersect a smaller cell, is equal to 2^k [*See Section 4.5.7*].

3. The factor $(2^{12^k} + 1)^2$ corresponds to the possibilities for the random bits on blue corner positions in these two quarters. Let us recall that the number of such positions in a quarter is equal to 12^k. Thus 2^{12^k} is the number of possibilities for the random bits when the frequency bit is 1, and 1 is the number of possibilities when the frequency bit is 0. [*See Section 4.5.2 and Section 4.5.7*].

4. The factor $2^{2.12^k s_k}$ corresponds to the possibilities for random bits in the computation quarters, since the frequency bit is determined to be s_k [*See Section 4.5.2 and Section 4.5.7*].

5. The last factor $2^{2.4.(4^k - 1)}$ corresponds to the number of possibilities for the undetermined `in`, `out` symbols in the two simulation quarters. Let us recall that $4^k - 1$ is the number of lines (resp. columns) intersecting a quarter of an order k cell (since the number of non determined symbols correspond the the number of possible contacts a segment can have with the reticle or the walls of the cell). [*See Section 4.5.6*].

6. The factor 4 is the number of sides of a quarter and 2 is the number of simulation quarters.

7. κ_k^* denotes the number of possibilities for the set of symbols in the error signals sublayer. This number is not simple to express, since it depends on the states of machine heads on the sides of the area after computation. However, for this reason it can be computed. We have $\kappa_k^* \leq 4^{6.(4^{k+1}+1)}$ for the reason that the possible sets of error symbols correspond to the choices of at most four symbols on every position of the walls and the reticle. [*See Section 4.5.7*].

When $k \leq N$, this number is

$$2^{4.12^k s_k},$$

which corresponds to the number of possible sets of random bits.

Upper and lower bounds:

Let $m > N$. We shall give an upper bound and a lower bound on the number of $(2n+1)$-blocks in the language of $X_{s,N}$, for all n. This depends on the integer m.

The **lower bound** is obtained as follows:

1. we give a lower bound on the number of blue corners in an order $k \leq m$ cells, included in some translate of $[\![-n,n]\!]^2$.

2. we give a lower bound on the number of non-blue corners in this set.

3. taking the product of the possible patterns over the union of these cells, we get a lower bound.

Consider some configuration x. The set $[\![-n,n]\!]^2$ contains at least $\left(\lfloor \frac{2n+1}{2.4^{m+1}} \rfloor - 1\right)^2$ translates of $[\![1, 2.4^{m+1}]\!]^2$ centered on an order m cell. Moreover, there are at least a number $\left(\lfloor \frac{2n+1}{2} \rfloor - 1\right)^2$ of translates of $[\![0,1]\!]^2$ such that the $(0,0)$ symbol is a blue corner. On each of these squares, the number of possible patterns on the set $[\![0,1]\!]^2 \backslash (0,0)$ is $2^{\lfloor 4h \rfloor}$.

$$\begin{array}{rcl} N_{2n+1}(X_{s,N}) & \geq & \prod_{k=N+1}^{m} \left(4.|\mathcal{A}|^{2.2^k} |\mathcal{Q}|^{6.2^k}.(2^{12^k}+1)^2.2^{2.12^k s_k}.2^{2.4.(4^k-1)}.\kappa_k^*\right)^{\left(\lfloor \frac{2n+1}{2.4^{k+1}} \rfloor - 1\right)^2} \\ & & \cdot \prod_{k=1}^{N} \left(2^{\left(\lfloor \frac{2n+1}{2.4^{k+1}} \rfloor -1\right)^2 .4.12^k s_k}\right).2^{\left(\lfloor \frac{2n+1}{2} \rfloor -1\right)^2 \lfloor 4h \rfloor} \end{array}.$$

As a consequence,

$$\begin{array}{rcl} \frac{\log_2(N_{2n+1}(X_{s,N}))}{(2n+1)^2} & \geq & \frac{1}{(2n+1)^2} \sum_{k=N+1}^{m} \left(\lfloor \frac{2n+1}{2.4^{k+1}} \rfloor - 1\right)^2 (2 + 2.2^k. \log_2(|\mathcal{A}|) + 6.2^k. \log_2(|\mathcal{Q}|) + 8.(4^k-1) \\ & & + \log_2(\kappa_k^*)) + \frac{1}{(2n+1)^2} \sum_{k=N+1}^{m} \left(\lfloor \frac{2n+1}{2.4^{k+1}} \rfloor - 1\right)^2 \left(2.12^k(1+s_k)\right) \\ & & + \frac{1}{(2n+1)^2} \sum_{k=1}^{N} \left(\lfloor \frac{2n+1}{2.4^{k+1}} \rfloor - 1\right)^2 .4.12^k s_k \\ & & + \frac{1}{(2n+1)^2} \cdot \left(\lfloor \frac{2n+1}{2} \rfloor - 1\right)^2 \lfloor 4h \rfloor \end{array}.$$

Taking n tending towards infinity, we obtain:

$$\begin{array}{rcl} h(X_{s,N}) & \geq & \sum_{k=N+1}^{m} \left(\frac{1}{32.16^k} + \frac{1}{32.8^k} \log_2(|\mathcal{A}|) + \frac{3}{32.8^k} \log_2(|\mathcal{Q}|) + \frac{1}{8.4^k} + \frac{\log_2(\kappa_k^*)}{4.16^{k+1}}\right) \\ & & + \frac{1}{2} \sum_{k=N+1}^{m} \frac{3^k}{4^{k+2}}(1+s_k) \\ & & + \sum_{k=0}^{N} \frac{3^k}{4^{k+2}} s_k \\ & & + \frac{\lfloor 4h \rfloor}{4} \end{array}.$$

Indeed, from the upper bound on $\kappa'(k)$, $k > N$ the series corresponding to this sequence in the formula above converges.

As this is true for all $m > N$, taking $m \to +\infty$, we obtain

$$\begin{array}{rcl} h(X_{s,N}) & \geq & \sum_{k=N+1}^{+\infty} \left(\frac{1}{32.16^k} + \frac{1}{32.8^k} \log_2(|\mathcal{A}|) + \frac{3}{32.8^k} \log_2(|\mathcal{Q}|) + \frac{1}{8.4^k} + \frac{\log_2(\kappa_k^*)}{4.16^{k+1}}\right) \\ & & + \frac{1}{2} \sum_{k=N+1}^{+\infty} \frac{3^k}{4^{k+2}}(1+s_k) \\ & & + \sum_{k=0}^{N} \frac{3^k}{4^{k+2}} s_k \\ & & + \frac{\lfloor 4h \rfloor}{4} \end{array}.$$

On the other hand, the **upper bound** is obtained as follows:

1. Since any n-block of X_{adR} can be extended into an order $\lceil \log_2(n) \rceil + 4$ supertile, there exists some $K > 0$ such that for all n the number of n-blocks in the structure layer is smaller than $(2^{K(\lceil \log_2(n)\rceil+4)})^2 \leq 2^{9K} n^{2K}$.

2. For all $m \leq n$ and x, the set $[\![-n,n]\!]^2$ is covered by at most $\left(\lceil \frac{2n+1}{2.4^{m+1}} \rceil + 1\right)^2$ translates of $[\![1, 2.4^{m+1}]\!]^2$ centered on an order m cell.

3. The set $[\![-n,n]\!]^2$ is covered by at most a number $\left(\lceil \frac{2n+1}{2} \rceil + 1\right)^2$ of translates of $[\![0,1]\!]^2$ such that the $(0,0)$ symbol is a blue corner. On each of these squares, the number of possible patterns on the set $[\![0,1]\!]^2 \backslash (0,0)$ is $2^{\lfloor 4h \rfloor}$.

4. $\Lambda'_m(x)$ denotes the set of positions that are not in an order $\leq m$ cell in x. Then the number of possibilities for these positions in $[\![-n,n]\!]^2$ is smaller than $c^{|\Lambda'_n(x) \cap [\![-n,n]\!]^2|}$, where c is the cardinality of the alphabet of the subshift $X_{s,N}$.

Hence we have the following inequality:

$$\begin{array}{rcl} N_{2n+1}(X_{s,N}) & \leq & 2^{9K} n^{2K} . \prod_{k=N+1}^{m} \left(4.|\mathcal{A}|^{2.2^k} |\mathcal{Q}|^{6.2^k} .(2^{12^k}+1)^2 . 2^{2.12^k s_k} . 2^{2.4.(4^k-1)} . \kappa_k^*\right)^{\left(\lceil \frac{2n+1}{2.4^{k+1}} \rceil + 1\right)^2} \\ & & . \prod_{k=0}^{N} \left(2^{\left(\lceil \frac{2n+1}{2.4^{k+1}} \rceil + 1\right)^2 . 4.12^k s_k}\right) . 2^{\left(\lceil \frac{2n+1}{2} \rceil + 1\right)^2 \lfloor 4h \rfloor} . c^{|\Lambda'_m(x) \cap [\![-n,n]\!]^2|} \end{array} .$$

This implies, as for the lower bound, and since $\log_2(2^{9K} n^{2K})/(2n+1)^2 \to 0$ when n tends to infinity, and from the third point of Lemma 4.32, that

$$\begin{array}{rcl} h(X_{s,N}) & \leq & \sum_{k=N+1}^{+\infty} \left(\frac{1}{32.16^k} + \frac{1}{32.8^k} \log_2(|\mathcal{A}|) + \frac{3}{32.8^k} \log_2(|\mathcal{Q}|) + \frac{1}{8.4^k} + \frac{\log_2(\kappa_k^*)}{4.16^{k+1}}\right) \\ & & + \frac{1}{2} \sum_{k=N+1}^{+\infty} \frac{3^k}{4^{k+2}} (1+s_k) \\ & & + \sum_{k=0}^{N} \frac{3^k}{4^{k+2}} s_k \\ & & + \frac{\lfloor 4h \rfloor}{4} + \log_2(c) \frac{3^{m+1}}{4^{m+2}} \end{array} .$$

Taking $m \to +\infty$, as $3^m/4^m \to 0$, we have the upper bound:

$$\begin{array}{rcl} h(X_{s,N}) & \leq & \sum_{k=N+1}^{+\infty} \left(\frac{1}{32.16^k} + \frac{1}{32.8^k} \log_2(|\mathcal{A}|) + \frac{3}{32.8^k} \log_2(|\mathcal{Q}|) + \frac{1}{8.4^k} + \frac{\log_2(\kappa_k^*)}{4.16^{k+1}}\right) \\ & & + \frac{1}{2} \sum_{k=N+1}^{+\infty} \frac{3^k}{4^{k+2}} (1+s_k) \\ & & + \sum_{k=0}^{N} \frac{3^k}{4^{k+2}} s_k \\ & & + \frac{\lfloor 4h \rfloor}{4} \end{array} .$$

□

4.6.3 Choosing the parameters values

In this section we explain how to choose s, N and M such that the entropy of $X_{s,N}$ is equal to h and there exists some M such that for all $k \geq 1$ integer, $s_{2kM} = 1$ and $s_{(2k+1)M} = 0$. This constraint on the sequence s will serve for the linear net gluing property. Here is the process that we follow for these choices:

1. we choose $N > 0$ and $M > 0$ such that

$$h > \sum_{k=N+1}^{+\infty} \kappa_k + \sum_{k=1}^{+\infty} \frac{3^{2kM}}{4^{2(kM+1)}} + \frac{\lfloor 4h \rfloor}{4}.$$

This is possible since $\frac{\lfloor 4h \rfloor}{4} < h$. The other terms of the sum in the left member of this inequality tends to 0 when N, M tends to $+\infty$. For this value of M, we impose the constraint on s that for all $k \geq 1$ integer, $s_{2kM} = 1$ and $s_{(2k+1)M} = 0$.

2. Since

$$\sum_{k=0} \frac{3^k}{4^{k+2}} + \frac{\lfloor 4h \rfloor}{4} = \frac{1}{4} + \frac{\lfloor 4h \rfloor}{4} > h,$$

the maximal value for the entropy $h(X_{s,N})$ for s verifying the constraint is greater than h. Moreover, the minimal value is smaller than h. The number

$$z' = \sum_{k \geq N+1} \kappa_k + \sum_{k=1}^{+\infty} \frac{3^{2kM}}{4^{2(kM+1)}} + \frac{\lfloor 4h \rfloor}{4}$$

is computable, and as a consequence the number

$$z = h - z'$$

is a Π_1-computable number. Hence we now have to choose s a Π_1-computable sequence in $\{0,1\}^{\mathbb{N}}$ such that the number

$$\frac{1}{2} \sum_{k=N+1}^{+\infty} \frac{3^k}{4^{k+2}} s_k + \sum_{k=0}^{N} \frac{3^k}{4^{k+2}} s_k$$

is equal to z. This is possible since z is a Π_1-computable number.

For these values of s and N and given the choice of M, the subshift $X_{s,N}$ is denoted X_h.

4.7 Linear net gluing of X_h

In this section we prove that the subshift X_h is linearly net gluing. This proof consists of two steps: first proving that any block can be extended into a cell, with control over the size of this cell. Then we prove the gluing property on cells having the same order. For reading this section, the reader should have some familiarity with the construction of the Robinson subshift [Rob71], presented in Chapter 3.

4.7.1 Completing blocks

The point of this section is to prove the following lemma:

Lemma 4.35. *Let $n \geq 0$ an integer, and P some $2^{n+1}-1$-block. This pattern can be completed into an admissible pattern in X_h over an order*

$$\left(2\left(\left\lceil \frac{n+1}{2.2M} \right\rceil + 1\right) + 3\right) M$$

cell.

Proof. Let $n \geq 0$ an integer and P some $2^{n+1}-1$-block which appears in some configuration x of the subshift X_h.

1. **Intersecting four order $2n$ supertiles:**

   This part of the proof is similar to the beginning of the proof of Lemma 3.28 in Chapter 3. The pattern P can be extended in the configuration x into a pattern over some pattern over one of the structure layer patterns on Figure 4.40, Figure 4.41, and Figure 4.42, composed with four $2n$ order supertiles with a cross separating them. These figures are not a strict reproduction of the ones in Chapter 3: we add possible colorations of the quarters included in the describtion of the structure layer in this construction. Moreover, the coloration of the reticle induce two possibilities to color some patterns: hence the types 4 and 10 are doubled by type $4'$ and $10'$ here.

2. **Completing the structure, frequency bits and basis:**

   All these patterns intersect non-trivially at most two different cells in the configuration x, one included into the other. The intersection with the small one is included into the union of two quarters of the cell with the separating segment. The intersection with the great one is included into a quarter of this cell. Similarly as in the proof of Lemma 3.28, the new formed pattern can be completed into an order $\left\lceil \frac{\left\lceil \log_2(2^{n+1}-1)\right\rceil}{2} \right\rceil + 2 + k$ cell, for any $k \geq 0$ in the Robinson sub-layer. However, in the present proof, we have to take care about the frequency bits of these two cells. Indeed, the order of the cell into which the pattern is completed has to be coherent with the frequency bit.

   That is why we slightly modify the way we complete the structure layer. In the cases when the pattern intersects non-trivially two cells (the schemata corresponding to the case of an intersecting with two cells are the numbers $1, 2, 3$, the second $5'$, $6, 7$, the third 8, and the first 9 on Figure 4.40, Figure 4.41, and Figure 4.42), we first complete the smallest cell into a cell having minimal order greater than $\left\lceil \frac{\left\lceil \log_2(2^{n+1}-1)\right\rceil}{2} \right\rceil + 2$. This is done in such a way that the corresponding bit is imposed to be equal to the actual bit attached to the part of the cell intersecting the pattern at this point. This means that we extend this part into an order $k.M$ cell, with k equal to

   $$2\left(\left\lceil \frac{\left\lceil \log_2(2^{n+1}-1)\right\rceil}{2.2M} \right\rceil + 1\right)$$

   or

   $$2\left(\left\lceil \frac{\left\lceil \log_2(2^{n+1}-1)\right\rceil}{2.2M} \right\rceil + 1\right) + 1$$

   Then the part of the second cell is completed into an order $k'.M$ cell, with k' equal to

   $$2\left(\left\lceil \frac{\left\lceil \log_2(2^{n+1}-1)\right\rceil}{2.2M} \right\rceil + 1\right) + 2$$

   or

   $$2\left(\left\lceil \frac{\left\lceil \log_2(2^{n+1}-1)\right\rceil}{2.2M} \right\rceil + 1\right) + 3.$$

   Thus, in any case, the initial pattern can be completed into an order

   $$\left(2\left(\left\lceil \frac{n+1}{2.2M} \right\rceil + 1\right) + 3\right) M$$

cell.

The case when the pattern intersects non-trivially only one cell, this completion is done similarly.

After this, one can complete the frequency bits layer according to the values in the initial pattern. One can also complete the synchronization net sublayer and the synchronization layer, since these are determined by the Robinson sublayer and the frequency bits. Then the random bits are chosed according to the frequency bits.

3. **Completing the computation areas, machines trajectories, and error signals:**

   In this paragraph, we describe how to complete the other layers (computation areas and machines layers) over this completed pattern.

   For each of the two non-trivially intersected cells, the lines and columns that do not intersect the initial pattern and that are between this pattern and the nucleus are colored $(\texttt{out}, \texttt{out})$. This allows the extension of the pattern in the machine layer simply by transport information between the nucleus and this part of the area. Indeed, there is not computation position outside the initial pattern. This is illustrated on Figure 4.33. When completing the machine's trajectories in the direction of time, we simply apply the computation rules of the machine.

   Error signals for the computation areas are triggered by these choices. We choose the propagation direction according to the presence or the absence of an error signal in the initial pattern. This is illustrated on Figure 4.32. The empty tape and sides signals and error signals of the machines are completed in a similar way.

   If the nucleus was in the initial pattern, then all these layers can be completed according to the configuration x. If this was not the case, then we choose the DNA such that the quarters present in the initial pattern - there are at most two since the nucleus is not in the initial pattern - are not represented in the DNA.

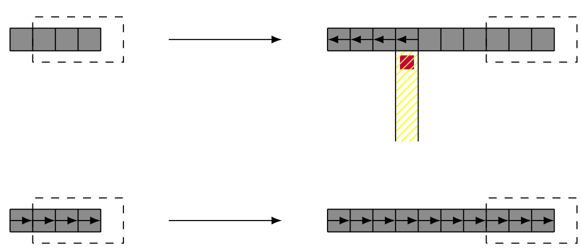


Figure 4.32: Illustration of the completion of the arrows according to the error signal in the known part of the area, designated by a dashed rectangle. The yellow color designates some $(\texttt{out}, \texttt{out})$ column.

□

4.7.2 Linear net gluing of X_h

Theorem 4.36. *The subshift X_h is linearly net gluing.*

Proof. Let P and Q be two $(2^{n+1}-1)$-blocks in the language of X_h. These two patterns can be completed into admissible patterns over order

$$\left(2\left(\left\lceil\frac{n+1}{2.2M}\right\rceil+1\right)+3\right)M$$

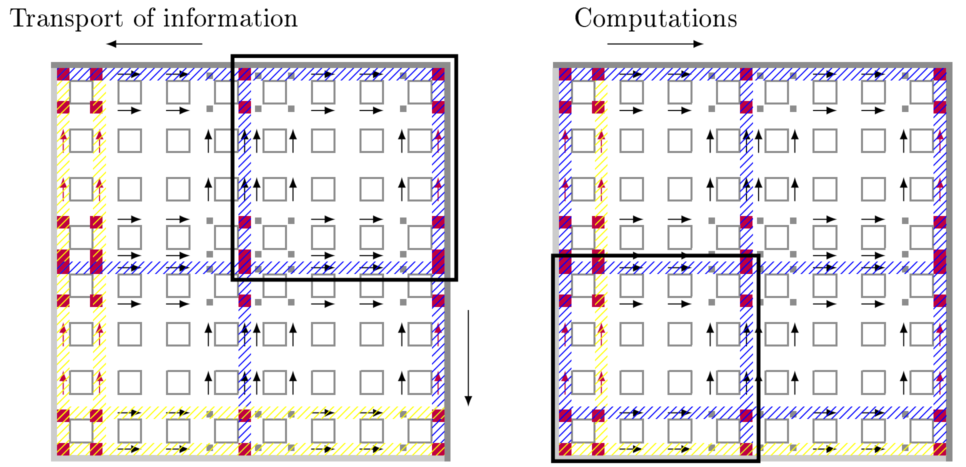


Figure 4.33: Illustration of the completion of the `in`, `out` signals and the space-time diagram of the machine. The known part of the cell is surrounded by a black square. The yellow lines and columns are colored with a symbol different from $(\mathtt{in}, \mathtt{in})$ in the computation areas layer, while the blue columns and lines are.

cells. The gluing set of these two cells is

$$4^{\left(2\left(\left\lceil \frac{n+1}{2.2M} \right\rceil +1\right)+3\right)M+2}.\mathbb{Z}^2 \backslash (0,0).$$

This means that there exists some vector $\mathbf{u}$ such that the gluing set of the pattern P contains

$$\mathbf{u} + (2^{n+1} - 1 + f(2^{n+1} - 1)).\mathbb{Z}^2 \backslash (0,0),$$

where

$$f(2^{n+1} - 1) = 4^{\left(2\left(\left\lceil \frac{n+1}{2.2M} \right\rceil +1\right)+3\right)M+2} - 2^{n+1} + 1 = O(2^{n+1} - 1).$$

As a consequence of Proposition 4.5, the subshift X_h is linearly net gluing. □

4.8 Transformation of X_h into a linearly block gluing SFT

In this section, we prove that every Π_1-computable non-negative real number is the entropy of some linearly block gluing $\mathbb{Z}^2$-SFT.

In order to prove this assertion, we use modified versions of the operator $d_{\mathcal{A}}$, depending on an integer parameter $r \geq 1$, that we denote $d_{\mathcal{A}}^{(r)}$. The definition of the operators $d_{\mathcal{A}}^{(r)}$ consists in imposing that the curves appearing in the definition of $d_{\mathcal{A}}$ are composed by length r straight segments, as illustrated on Figure 4.34.

Each of these segments can be superimposed with random colors defined to be a length r word amongst the words $0^r, 10^{r-1}, ..., 1^r$. This sequence is imposed to be 0^r when the segment is not surrounded with other segments, as illustrated on Figure 4.35.

The idea behind this definition is that with these random colors, the patterns crossed only by straight curves are the most numerous ones, and thus the entropy is easier to compute, with the cost of a parasitic entropy. The parameter r serves to control the parasitic entropy.

$$\begin{array}{ccccccccc}
\rightarrow & \rightarrow & \rightarrow & \downarrow & \downarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow \\
\downarrow & \downarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow
\end{array}$$

Figure 4.34: Making the curves more rigid. In this example, $r = 3$.

$$\begin{array}{ccccccccc}
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \\
\rightarrow & \rightarrow & \rightarrow & \color{red}{\rightarrow} & \color{red}{\rightarrow} & \color{red}{\rightarrow} & \rightarrow & \rightarrow & \rightarrow \\
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow
\end{array}$$

Figure 4.35: Segment (colored red) surrounded by other ones. In this example, $r = 3$.

$\tilde{\mathcal{A}}_r$ denotes the alphabet of $d^{(r)}_{\mathcal{A}}$. The operators $d^{(r)}_{\tilde{\mathcal{A}}_r} \circ \rho \circ d^{(r)}_{\mathcal{A}}$ still transform linearly net gluing subshifts into linearly block gluing ones, and the entropy of $d^{(r)}_{\mathcal{A}}(Z)$ for a subshift Z is a function of $h(Z)$:

Theorem 4.37. *For any Z on $\mathbb{Z}^2$ and on alphabet $\mathcal{A}$, the entropy of $d^{(r)}_{\mathcal{A}}(Z)$ is*

$$\boxed{h(d^{(r)}_{\mathcal{A}}(Z)) = h(Z) + \frac{\log_2(1+r)}{r}}.$$

Since any non-negative Π_1-computable number is the entropy of a linearly net gluing SFT, for all $r \geq 1$, all the numbers in $\left[2\frac{\log_2(1+r)}{r}, +\infty\right[$ (the factor 2 here comes from the fact that we apply two operators) are entropy of a linearly block gluing SFT. As a consequence, since 0 is the entropy of the full shift on alphabet $\{0\}$, which is linearly block gluing, all the non-negative Π_1-computable numbers are entropy of a linearly block gluing SFT.

Question 2. *In this proof, the entropy 0 is obtained in a different way than other Π_1-computable numbers. Is there some non-trivial SFT which is linearly block and have entropy 0 ?*

4.8.1 Definition of the operators $d^{(r)}_{\mathcal{A}}(Z)$

Let Z be a $\mathbb{Z}^2$-SFT on alphabet $\mathcal{A}$, and $r \geq 1$. The subshift $d^{(r)}_{\mathcal{A}}(Z)$ is defined as a product of four layers:

1. the first layer is Δ,
2. the second one is Z, with similar rules with respect to Δ as in the definition of $d_{\mathcal{A}}$.
3. the counter layer [Section 4.8.1.1]: in this layer we impose, using a counter, the curves to be composed of length r segments.
4. the random colors layer [Section 4.8.1.2]: here we superimpose random colors to the length r segments of curves.

4.8.1.1 Counter layer

Symbols:

The elements of $\mathbb{Z}/r\mathbb{Z}$ and a blank symbol.

Local rules:

- **Localization:** the non blank-symbols are superimposed on and only on the positions having a $\rightarrow$ symbol in the Δ layer.

- **Incrementing the counter:** over a pattern

$$\rightarrow\rightarrow$$

or

$$\begin{array}{cc} \rightarrow & \downarrow \\ \downarrow & \rightarrow \end{array}$$

in the Δ layer, if the value of the counter on the left position with $\rightarrow$ is $\bar{i}$, then the value on the right position is $\bar{i}+\bar{1}$.

- **The curves can shift downwards only on position with maximal counter value:** on the pattern

$$\begin{array}{cc} \rightarrow & \downarrow \\ \downarrow & \rightarrow \end{array},$$

the left position with $\rightarrow$ has counter value equal to $\overline{r-1}$.

Global behavior:

On each curve induced by the restriction to the Δ layer we superimpose independent counters that are incremented along the curves. A curve can shift downwards only when the counter has maximal value. This implies that the curves are composed of length r segments. A segment is defined to be a part of the curve between two consecutive positions where the counter has value $\overline{0}$ and $\overline{r-1}$.

4.8.1.2 Random colors layer

Symbols:

The symbols of this layer are $0, 1$ and a blank symbol.

Local rules:

- **Localization:** the bits $0, 1$ are superimposed on and only on positions with symbol $\rightarrow$.

- **Restriction of the possible colors:** Along a length r segment of curve, the symbol 1 propagates to the left. The symbol 0 propagates to the right.

- **Isolation rule:** if a segment is not surrounded by other segments, its color is 0^r.

Global behavior:

Each length r segment is attached with a word in $\{0^r, 10^{r-1}, ..., 1^r\}$. Moreover, if the segment is not surrounded by other segments, its color is 0^r.

4.8.2 Transformation of linearly net gluing subshifts into linearly block gluing ones

We have a result similar to Theorem 4.26 for the operators $d_{\mathcal{A}}^{(r)}$:

Theorem 4.38. *For all $r \geq 0$, the operator $d_{\tilde{\mathcal{A}}_r}^{(r)} \circ \rho \circ d_{\mathcal{A}}^{(r)}$ transforms linear net-gluing subshifts of finite type into linear block gluing ones.*

Proof. The steps of the proof of this theorem are exactly the same ones as for the proof of Theorem 4.26. However, the gap function of the image subshift is r times the gap function of the image subshift obtained when applying $d_{\tilde{\mathcal{A}}} \circ \rho \circ d_{\mathcal{A}}$. The presence of the counters do not have any impact since when we proved that two patterns can be glued we don't connect the curves of the two patterns. □

Δ'_r denotes the subshift that consists in the product of the Δ layer with the counter layer and random colors layer, with rules relating these layers. Moreover, Δ_r denotes the product of the Δ layer with the counter layer, with rules relating these layers.

The following sections are devoted to the proof of Theorem 4.37.

4.8.3 Lower bound

Lemma 4.39. *For any subshift Z on alphabet $\mathcal{A}$, we have the following lower bound on the entropy of $d_{\mathcal{A}}^{(r)}(Z)$:*

$$h(d_{\mathcal{A}}^{(r)}(Z)) \geq h(Z) + \frac{\log_2(1+r)}{r}.$$

Proof. The language of $d_{\mathcal{A}}^{(r)}(Z)$ contains all the kr-blocks whose symbols in the Δ layer are all equal to $\rightarrow$ and such that in the first column, the value of the counter is $\overline{0}$ in each line.

The number of such patterns is $N_{kr}(Z).(r+1)^{rk^2}$ (the first factor is the number of choices in the Z layer and the other one is the number of choices in the random colors layer). Hence

$$N_{kr}(d_{\mathcal{A}}^{(r)}(Z)) \geq (r+1)^{rk^2}.N_{kr}(Z).$$

Which implies

$$\frac{\log_2(N_{kr}(d_{\mathcal{A}}^{(r)}(Z)))}{kr.kr} \geq \frac{\log_2(1+r)}{r} + \frac{\log_2(N_{kr}(Z))}{kr.kr}.$$

We deduce, taking $k \rightarrow +\infty$, that

$$h(d_{\mathcal{A}}^{(r)}(Z)) \geq \frac{\log_2(1+r)}{r} + h(Z).$$

□

4.8.4 Upper bound

We prove an upper bound for $h(d_{\mathcal{A}}^{(r)}(Z))$ in two steps:

1. In Section 4.8.4.1, we prove a bound on the number of possible pseudo-projections of a kr-block onto the subshift Z. This is done ussing an upper bound on the number of curves crossing a n-block.
2. In Section 4.8.4.2, we give an upper bound on the number of kr-blocks in Δ'_r. We do this by analyzing the possible ways to extend a kr-block in the language of this subshift into a $(k+1)r$-block which stays admissible.

4.8.4.1 Upper bound on the number of pseudo-projections

Lemma 4.40. *Let $n \geq 1$ and P be some n-block in the language of Δ_r. The number of curves crossing P is equal to $k_n + k'_n$, where k_n is the number of $\rightarrow$ symbols on the south west - north east diagonal and k'_n is the number of patterns*

$$\begin{array}{cc} \rightarrow & \downarrow \\ & \rightarrow \end{array}$$

such that the $\downarrow$ symbol is on the south west - north east diagonal.

Proof. Each curve crossing P crosses the diagonal. This is due to the fact that in each column, the curve goes straightly onto the right or is shifted downwards. When it crosses the diagonal, there are two possibilities: either it is shifted downwards, and it corresponds to the pattern

$$\begin{array}{cc} \rightarrow & \downarrow \\ & \rightarrow \end{array},$$

or it is not and this corresponds to the symbol $\rightarrow$ on the diagonal. Hence counting this patterns gives the number of curves crossing P. On Figure 4.36, the pattern has three times the symbol $\rightarrow$ on the diagonal and once the pattern

$$\begin{array}{cc} \rightarrow & \downarrow \\ & \rightarrow \end{array}.$$

One can see that the total is equal to the number of curves crossing this pattern.

$$\begin{array}{cccccc}
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \color{red}{\rightarrow} \\
\rightarrow & \rightarrow & \rightarrow & \color{red}{\rightarrow} & \color{red}{\downarrow} & \downarrow \\
\rightarrow & \downarrow & \downarrow & \downarrow & \color{red}{\rightarrow} & \rightarrow \\
\downarrow & \rightarrow & \color{red}{\rightarrow} & \rightarrow & \downarrow & \downarrow \\
\rightarrow & \color{red}{\rightarrow} & \rightarrow & \downarrow & \rightarrow & \rightarrow \\
\downarrow & \downarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow
\end{array}$$

Figure 4.36: Example of pattern P in the Δ layer. The patterns on the diagonal allow the number of curves to be counted. In this example, $r = 3$.

□

Lemma 4.41. *The number of pseudo-projections of a kr-block pattern in the language of $d_{\mathcal{A}}^{(r)}(Z))$ is smaller or equal to $N_{kr}(Z)$.*

Proof. A a consequence of Lemma 4.40, a kr-block contains at most kr curves and the number of positions of a curve in a kr block is smaller than kr. Hence the number of pseudo-projections of a kr-block on Z is smaller than $N_{kr}(Z)$. □

4.8.4.2 Upper bound on the number of colored curves patterns

In this section, we give an upper bound on the number of kr-block in the language of Δ'_r, for $k \geq 1$. It relies on an upper bound on the number of patterns in specific sets, defined as follows.

Consider some kr-block P in the language of Δ_r. Define $\mathbb{U}_P$ to be the minimal set containing $[\![0, kr-1]\!]^2$ such that there exists some pattern P' - which is unique - on support $\mathbb{U}_P$:

- whose restriction on $[\![0, kr-1]\!]^2$ is P,

- and such that the leftmost (resp. rightmost) position in any curve in P' crossing the left (resp. right) side of P has counter $\overline{0}$ (resp. $\overline{r-1}$).

On Figure 4.37, one can find some example of such completion of a block P into the pattern P'.

$$
\begin{array}{cccccccccc}
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \downarrow & & \\
 & \rightarrow & \rightarrow & \rightarrow & \downarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \\
 & & \downarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & & \\
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow \\
 & & \downarrow & \downarrow & \downarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \\
 & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & &
\end{array}
$$

$$
\begin{array}{cccccccccc}
\overline{0} & \overline{1} & \overline{2} & \overline{0} & \overline{1} & \overline{2} & & & & \\
 & \overline{0} & \overline{1} & \overline{2} & & & \overline{0} & \overline{1} & \overline{2} & \\
 & & & & \overline{0} & \overline{1} & \overline{2} & & & \\
\overline{0} & \overline{1} & \overline{2} & \overline{0} & \overline{1} & \overline{2} & & \overline{0} & \overline{1} & \overline{2} \\
 & & & & & & \overline{0} & \overline{1} & \overline{2} & \\
 & \overline{0} & \overline{1} & \overline{2} & \overline{0} & \overline{1} & \overline{2} & & &
\end{array}
$$

Figure 4.37: Illustration of the extension of a block by completing the segments, $r = 3$. The pattern P is represented by dark symbols, and is completed into the projection of P' by red ones.

Let us denote $\mathcal{T}_{kr}$ the set of patterns in the language of Δ'_r whose projection on Δ_r is P' for some kr-block P in the language of Δ_r.

Lemma 4.42. *For all $k \geq 1$, we have the following upper bound on the number of kr-blocks in the language of Δ'_r:*

$$|N_{kr}(\Delta'_r)| \leq \alpha(r).2^{\lambda(r).k}(r+1)^{rk^2},$$

where $\alpha(r), \lambda(r) > 0$ depend only on r.

Idea: *the idea of the proof is to get an upper bound on the cardinality of $\mathcal{T}_{kr}$ for all $k \geq 1$ considering the possible extensions of a pattern in $\mathcal{T}_{kr}$ into a pattern of $\mathcal{T}_{(k+1)r}$. We derive then an upper bound for the number of kr-blocks in the language of Δ'_r.*

Proof. 1. **Upper bound on the extensions of patterns in $\mathcal{T}_{kr}$ into patterns of $\mathcal{T}_{(k+1)r}$:**

Consider some pattern Q in $\mathcal{T}_{kr}$. We will first consider the number of possibilities to extend this pattern on the right side and then on the top, as illustrated on Figure 4.38.

(a) **Extensions on the right side:**

The restriction C of Q on the rightmost complete column $\{kr-1\} \times [\![0, kr-1]\!]$ (colored gray on Figure 4.38) is sub-pattern of a pattern C' over $\{kr-1\} \times [\![-1, kr]\!]$ or $\{kr-1\} \times [\![0, kr]\!]$ such that this pattern is the (vertical) concatenation of patterns

$$\begin{array}{c} \rightarrow \\ \vdots \\ \rightarrow \\ \downarrow \end{array} .$$

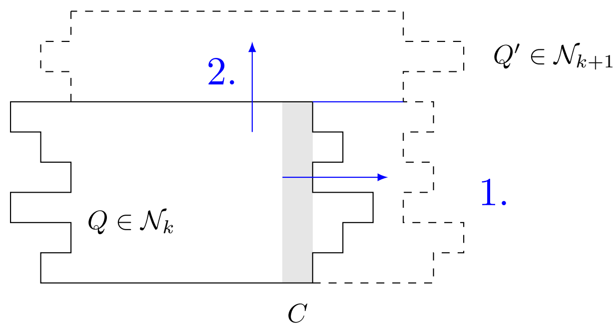


Figure 4.38: Illustration of the considered order for completion in order to give an upper bound on $|\mathcal{T}_{(k+1)r}|/|\mathcal{T}_{kr}|$.

One can see this by adding a symbol $\rightarrow$ on the top, and a symbol $\downarrow$ on the bottom if the bottommost symbol in this column is $\rightarrow$ (we do not complete C into C', which is just an artifact allowing an upper bound on the number of ways to extend Q on its right side). Each of these patterns corresponds to a set of vertically consecutive curves going out of the pattern Q through its right side.

A completion of Q on the right side is determined by the following choices:

- for each of the outgoing curves, choose if this curve is shifted downwards in the set of additional positions or not.
- for each of the added segments of curve, choose a color to superimpose over it.

Moreover, for each set of consecutive curves, if one of these curves is shifted downwards, this forces the curves in this set to be shifted downwards as well in the set additional positions. This shift is realized in a column on the left of the column where the first curve is shifted. Since the additional shift positions lie in a set of r consecutive columns, this means that only the r bottommost curves in this set can be shifted downwards in the additional columns. The number of possible choices for these shifts is $\min(j,r)$ (since the position of the shift is determined by the counter), where j is the number of curves in this set.

For this set of curves the number of choices for the colors is $(r+1)^j$. As a consequence, for a set of consecutive curves represented by a pattern

$$\begin{array}{c} \rightarrow \\ \vdots \\ \rightarrow \\ \downarrow \end{array},$$

with j symbols $\rightarrow$, the number of possible extensions on the right of this pattern is smaller than $(r+1)^{j+1}$. Since in this formula, $j+1$ is the height of the pattern

$$\begin{array}{c} \rightarrow \\ \vdots \\ \rightarrow \\ \downarrow \end{array},$$

the number of possible extensions of Q on the right side is smaller than the product of these numbers. This is equal to

$$(r+1)^{kr+2},$$

kr being the height of the pattern C.

(b) **Extensions on the top:**

Let us consider a possible completion $Q^{(1)}$ of Q on the right side. We give an upper bound on the number of possible ways to complete this pattern on the row $[\![0,(k+1)r-1]\!] \times \{kr\}$ just above this pattern. This depends only on the restriction of $Q^{(1)}$ on $[\![0,(k+1)r-1]\!] \times \{kr-1\}$. This way, taking the power r of this bound will provide a bound for the possible completions of $Q^{(1)}$ into a pattern $Q^{(2)}$ of $\mathcal{T}_{(k+1)r}$.

The restriction of $Q^{(1)}$ on its topmost row $[\![0,(k+1)r-1]\!] \times \{kr-1\}$ can be decomposed into a (possibly empty) concatenation of patterns

$$\rightarrow \dots \rightarrow \downarrow \dots \downarrow,$$

possibly followed on the right by a pattern

$$\rightarrow \dots \rightarrow,$$

and possibly preceded on the left by a pattern

$$\downarrow \dots \downarrow.$$

For instance, on the following pattern, we represent the decomposition with parentheses:

$$(\downarrow\downarrow\downarrow)\,(\rightarrow\rightarrow\rightarrow\downarrow\downarrow\downarrow)\,(\rightarrow\rightarrow\rightarrow\downarrow\downarrow\downarrow)\,(\rightarrow\rightarrow\rightarrow).$$

According to this decomposition, the possibilities for completing Q' on the top are as follows:

- If the pattern on the top row of $Q^{(1)}$ is equal to

  $$\rightarrow \dots \rightarrow,$$

  then the pattern can extended on top with some pattern which consists in the concatenation of

  $$\downarrow \dots \downarrow,$$

  with

  $$\rightarrow \dots \rightarrow$$

  on the right, one of which can be empty. For instance, if the pattern on the top row is

  $$\rightarrow \quad \rightarrow \quad \rightarrow \quad \rightarrow \quad \rightarrow \quad \rightarrow$$

  and $r = 3$, one can extend the pattern in the following ways:

  $$\begin{array}{cccccccc} \downarrow & \downarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & & \end{array},$$

  $$\begin{array}{cccccc} \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \end{array},$$

  or

  $$\begin{array}{cccccc} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow \end{array}.$$

- When we extend with the pattern

$$\to \ldots \to,$$

there are r possibilities for the positions of the counter symbols, and at most

$$(r+1)^{k+2}$$

possibilities for the random colors of the added segments (there are at most $(k+2)$ ones). As a consequence, in this case there are at most $r(r+1)^{k+2} \leq (r+1)^{k+3}$ possible extensions.

- When we extend with some pattern

$$\downarrow \ldots \downarrow \to \ldots \to,$$

there are r possibilities for the rightmost position of the row where the counter has value $\overline{0}$. For each j such that the added curve shifts downwards in a column between the jrth and the $(j+1)r-1$th one, the number of possible colorings of this curve is at most $(r+1)^{k-j+1}$ (since in this case, the number of added segments is at most $(k-j+1)$). As a consequence, the number of completions in this case is at most

$$r\sum_{j=1}^{k+1}(r+1)^j = r(r+1)\frac{(r+1)^{k+1}-1}{r+1-1} \leq (r+1)^{k+2}.$$

- When the pattern on the top row of $Q^{(1)}$ is

$$\downarrow \ldots \downarrow,$$

the only possibility for completion in the Δ layer is by

$$\to \ldots \to .$$

The number of possibilities in the other layers is given by the choice of the counter position and the colors, and is smaller than $(r+1)^{k+3}$.

- **Mixed cases:** For the same reasons as in the two previous cases, the number of possible extensions on the top of the $\to \ldots \to$ pattern in the decomposition of the top row pattern and the $\downarrow \ldots \downarrow$ pattern, and the leftmost (resp. rightmost) pattern

$$\to \ldots \to \downarrow \ldots \downarrow$$

when there is no pattern $\downarrow \ldots \downarrow$ (resp. $\to \ldots \to$) in the decomposition, are at most

$$(r+1)^{\left\lceil \frac{l}{r} \right\rceil + 3},$$

where l is the length of this pattern.
The possible extensions over the other patterns

$$\to \ldots \to \downarrow \ldots \downarrow$$

in the decomposition are as follows:

– the $r\left\lceil \frac{m}{r} \right\rceil$ rightmost symbols in the extending row are equal to $\rightarrow$, where m is the length of the sub-pattern $\downarrow \dots \downarrow$: this is imposed by the presence on the right of $\rightarrow$ symbols. This corresponds to a shifted curve, which has to go straight while there are symbols $\downarrow$ below it during a number rj of columns, for some j. In this step, the colors of the added segments is 0^r, since they are not surrounded by other segments. At this point, the completion over this part of the top row looks as follows:

$$\begin{array}{cccccccc} & & & & & \rightarrow & \rightarrow & \rightarrow \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \downarrow & \downarrow \end{array}.$$

– then we have to choose the position of the shift of this curve in the row that we are adding. Indeed, the top row pattern is preceded by $\downarrow$ symbols on the left. This means that this is shifted downwards there. The added curve has thus to be shifted in a column on the right of this one. For these added segments, we have to choose a color.

For these patterns, the number of possible extensions is thus at most

$$\sum_{k=0}^{\left\lfloor \frac{l}{r} \right\rfloor -1} (r+1)^k \leq (r+1)^{\left\lfloor \frac{l}{r} \right\rfloor},$$

where l is the length of this pattern. Indeed, there are at most $\left\lfloor \frac{l}{r} \right\rfloor$ added segment over this part of the top row, and that at least one of them has trivial color 0^r.
As a consequence, in these cases, the number of possible extensions over the top row is at most $(r+1)^{\frac{L}{r}+8}$, where L is the total length of the top row. As a consequence, since $L = rk$, this number is equal to

$$(r+1)^{k+9}.$$

In any of these cases the number of possible extensions is smaller than $(r+1)^{k+9}$. Hence the total number of possible extensions of a pattern in $\mathcal{T}_{kr}$ into a pattern in $\mathcal{T}_{(k+1)r}$ is at most $3(r+1)^{k+9}$, since there are three cases.

2. **Upper bound on the cardinality of $\mathcal{T}_{kr}$:**

As a consequence, the number of possible extensions of a pattern in $\mathcal{T}_{kr}$ into a pattern in $\mathcal{T}_{(k+1)r}$ is at most

$$(r+1)^{kr+2}\left(3(r+1)^{k+9}\right)^r = 3^r(r+1)^{2kr+9r+2}.$$

It follows, using inductively this inequality, that the number of pattern in $\mathcal{T}_{(k+1)r}$ is smaller than

$$|\mathcal{T}_r|.(r+1)^{k.(2+9r)}.3^{k.r}.(r+1)^{2r\sum_{i=1}^{k} i} = |\mathcal{T}_r|.(r+1)^{k.(2+9r)}.3^{kr}.(r+1)^{rk(k+1)}.$$

3. **Upper bound on $N_{kr}(\Delta'_r)$:**

As a consequence, the number of kr-blocks in the language of Δ'_r is smaller than

$$|\mathcal{T}_r|.(r+1)^{k.(2+10r)}.3^{kr}.(r+1)^{rk^2}.(r+1)^{2(k+1)r},$$

since any kr-block is sub-pattern of a pattern in $\mathcal{T}_{(k+1)r}$.

□

Proof. of Theorem 4.37: As a consequence of Lemma 4.41, and Lemma 4.42, we get that the number of kr-blocks in the language of $d_{\mathcal{A}}^{(r)}(Z)$ smaller than

$$N_{kr}(Z).\alpha(r).2^{\lambda(r).k}.(r+1)^{rk^2}.$$

Hence,

$$h(d_{\mathcal{A}}^{(r)}(Z)) \leq h(Z) + \frac{\log_2(1+r)}{r}$$

Using Lemma 4.39, we have the equality:

$$h(d_{\mathcal{A}}^{(r)}(Z)) = h(Z) + \frac{\log_2(1+r)}{r}.$$

□

Remark 33. *Although we don't prove it in this text, he construction realized in this chapter can be adapted to prove that known results about multidimensional SFT (existence of non-recursive points, existence of SFT with non-decidable language) can be adapted under the constraint of linear block gluing.*

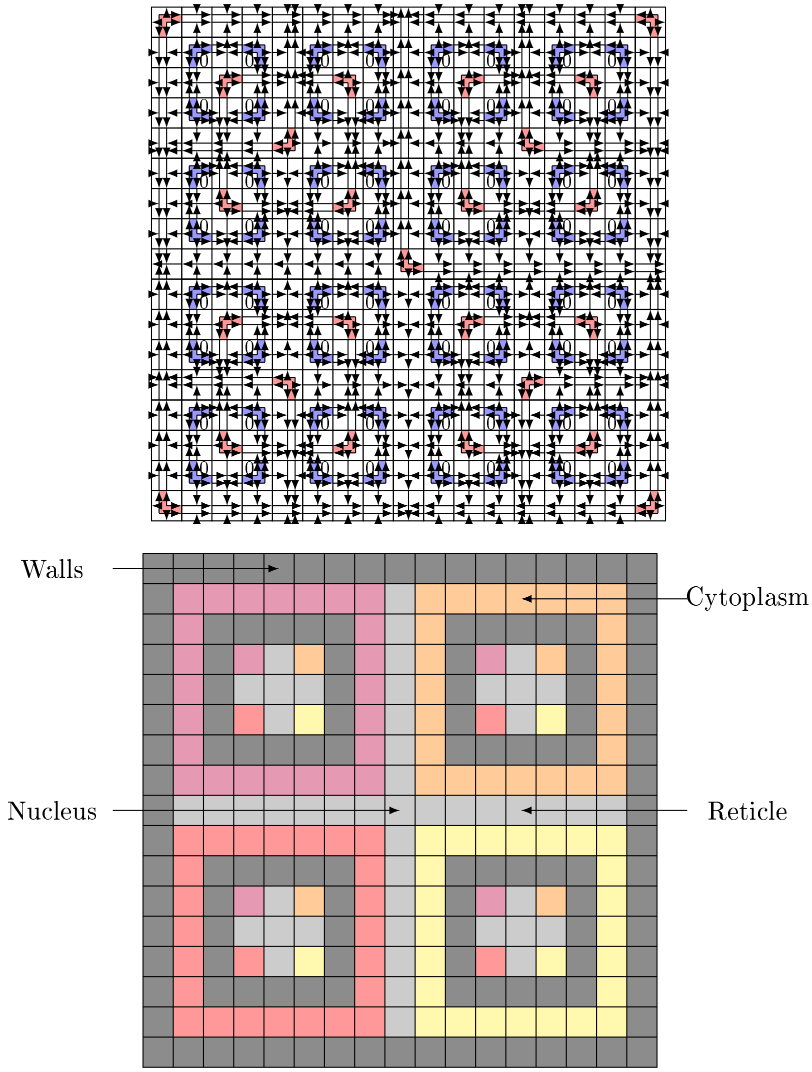


Figure 4.39: Example of the projection over the structure (first picture) and the synchronization (second one) layers of a pattern over an order 1 cell. This cell contains properly 4 order 0 cells, and 4.12 blue corners.

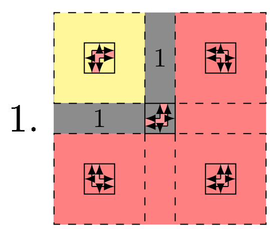


Figure 4.40: Possible orientations of four neighbor supertiles having the same order.

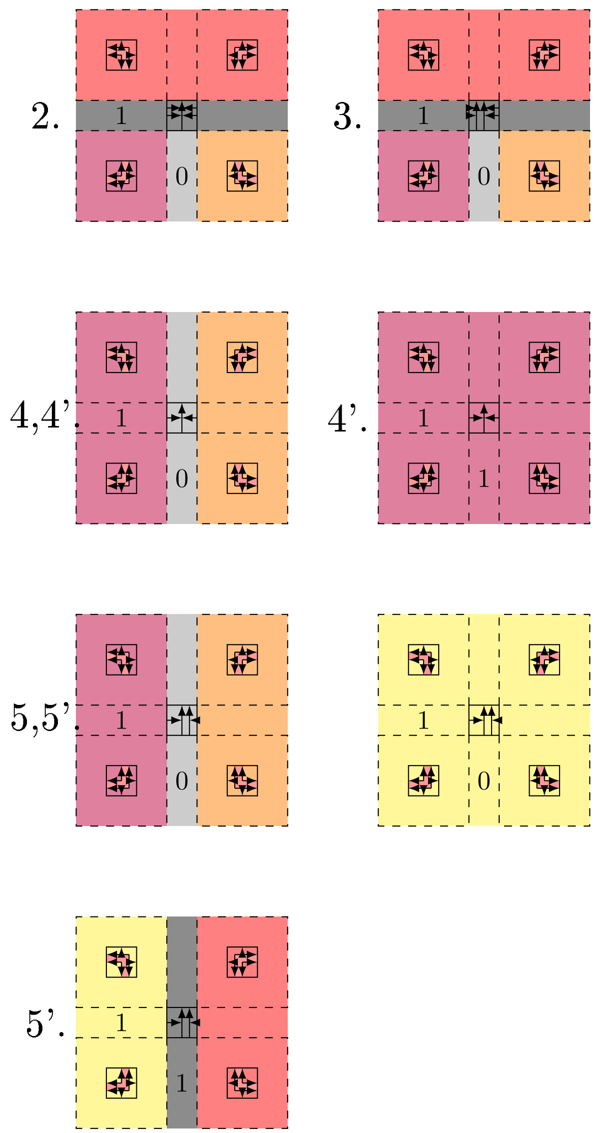


Figure 4.41: Possible orientations of four neighbor supertiles having the same order.

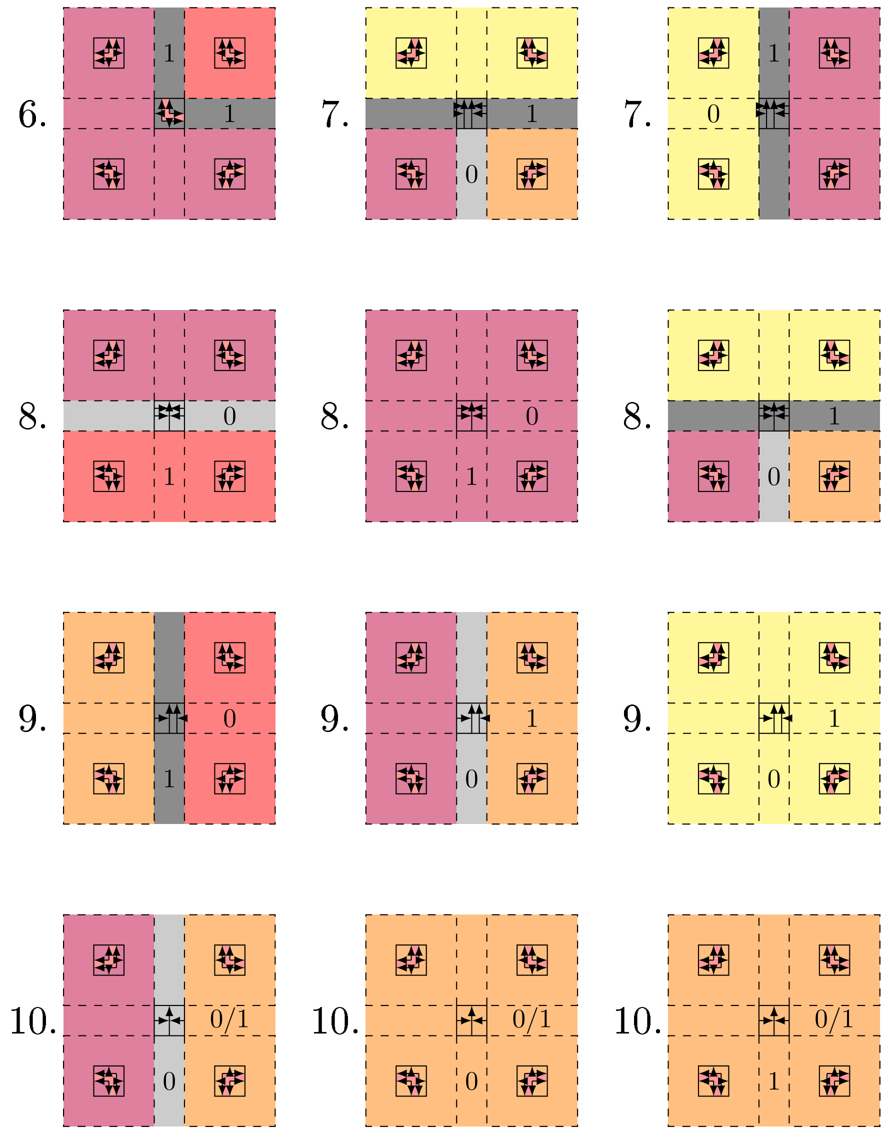


Figure 4.42: Possible orientations of four neighbor supertiles having the same order.

Chapter 5

Irreducibility with gap function: entropy of decidable subshifts

Sommaire

This chapter is a partial reproduction of the article [GH16].

Abstract

In this chapter, we introduce a notion of strong irreducibility with gap function. Then we exhibit a threshold condition for the gap function above which the possible entropies for decidable subshifts are all the Π_1-computable numbers, and under which the entropy is computable. This provides a more precise result than for SFTs, for which the same problem was studied in Chapter 4.

5.1 Introduction

The entropy of one dimensional subshifts of finite type (SFT) is well understood, since the language of a one dimensional SFT is accepted by a finite automaton. As a direct consequence, its entropy is a computable number. This is not the case for multidimensional subshifts, since all the Π_1-computable numbers are entropy of a multidimensional SFT. The reason of this difference seems to be that a unique dimension prevents the implementation of universal computation in SFT.

In Chapter 4, we studied the effect of block gluing with gap function on the computability of the entropy of multidimensional SFT. This seems that there is a threshold for the gap function above which the block gluing condition is not enough restrictive to prevent universal computation. All the

Π_1-computable numbers are entropy of SFT which verifies the condition. Under this threshold, the entropy becomes computable.

Exhibiting the threshold is hard, and the purpose of this chapter is to precise this for a more flexible class: decidable $\mathbb{Z}^d$-subshifts. We study them under an irreducibility condition, for which we introduce a gap function.

The situation is similar to the case of subshifts of finite type, except that we are able to precise the location of the threshold. Moreover, we this threshold corresponds to a summability condition.

More precisely, the main result presented in this chapter is the following one:

Theorem 1.3 (Theorem 5.9 and Theorem 5.12)**.** *Let $d \geq 1$ be an integer, and $f : \mathbb{N} \to \mathbb{N}$ a computable function.*

1. *If $\sum_{n=1}^{+\infty} \frac{f(n)}{n^2}$ converges at a computable rate, then there exists an algorithm that (uniformly) computes the entropy of f-irreducible decidable $\mathbb{Z}^d$-subshifts.*
2. *If $\sum_{n=1}^{+\infty} \frac{f(n)}{n^2} = +\infty$, then the set of numbers that are entropy of a decidable f-irreducible $\mathbb{Z}^d$-subshift are exactly the Π_1-computable numbers.*

In the following, we make the terms of this statement more precise. In particular, the class of functions such that $\sum_{n=1}^{+\infty} \frac{f(n)}{n^2}$ converges at a computable rate includes all the $O\left(\frac{n}{\log^{1+\alpha}(n+2)}\right)$ functions for α sufficiently small.

Results related to this question, including this one, are summed up in Table 1.

Subshift class	Irreducibility rate			
	None	$\geq$	$\leq$	Constant
SFT	Π_1-comput. [HM10]	?	computable †	computable [HM10]
$d \geq 2$	all Π_1-c. reals [HM10]	?	?	partial char. [PS15]
Decidable	Π_1-comput. [Sim06]	Π_1-comput. †	computable †	computable [Spa07]
$d \geq 1$	all Π_1-c. reals [HS08]	all Π_1-c. reals †	?	?

Table 5.1: (First line) Computational difficulty of computing the entropy; (Second line) Set of possible entropies. The symbols $\geq$ and $\leq$ designate rates respectively above and below the threshold; † symbols indicate the contribution presented in the present chapter.

5.2 Irreducibility with gap function and decidability

In this section, we provide specific definitions for the chapter. If X is a $\mathbb{Z}^d$-subshift on alphabet $\mathcal{A}$, $a \in \mathcal{A}$ and u a finite pattern, we denote $\#_a u$ the number of times that the letter a appears in the pattern u. We denote $\mathcal{A}^*$ the set of words on alphabet $\mathcal{A}$.

Let us recall that for two finite subsets $\mathbb{U}, \mathbb{V}$ of $\mathbb{Z}^d$, $\delta(\mathbb{U})$ is the diameter (for the maximum distance) of the set $\mathbb{U}$, and

$$\delta(\mathbb{U}, \mathbb{V}) = \min_{\mathbf{i} \in \mathbb{U}} \min_{\mathbf{j} \in \mathbb{V}} d(\mathbf{i}, \mathbf{j}).$$

5.2.1 Notion of irreducibility with gap function

Definition 5.1. *Let f be a function from $\mathbb{N}$ to $\mathbb{N}$ and X be a $\mathbb{Z}^d$-subshift. The subshift X is said to be **f-irreducible** when for any pair of finite subsets $\mathbb{U}, \mathbb{V}$ of $\mathbb{Z}^d$ such that*

$$\delta(\mathbb{U}, \mathbb{V}) \geq f(\max(\delta(\mathbb{U}), \delta(\mathbb{V})))$$

and u, v patterns on $\mathbb{U}, \mathbb{V}$ that are globally admissible for X, there exists a configuration x of X such that $x_{\mathbb{U}} = u$ and $x_{\mathbb{V}} = v$.

Remark 34. *If a subshift is f-irreducible (resp. f block gluing), and $g \geq f$, then the subshift is also g-irreducible (resp. g block gluing).*

Definition 5.2. *Consider $(u_n)_n$ some sequence of positive real numbers. The subshift X is said to be $O(u_n)$-irreducible if X is f-irreducible for some function $f : \mathbb{N} \to \mathbb{N}$ such that*

$$f(n) = O(u_n).$$

Remark 35. *The property of $O(1)$-irreducibility is usually called* ***strong irreducibility*** *in the literature. Moreover, the subshifts that are f-irreducible for some (non necessarily computable) function f are usually called* ***topologically mixing****. Definition 5.1 and Definition 5.2 extend these notions by mentioning the gap function of irreducibility.*

5.2.2 Equivalence of f-block gluing and f-irreducibility for $\mathbb{Z}$-subshifts

In this part, we present a technical proposition which allows us to derive the irreducibility property from the block gluing one.

For $\mathbb{U}, \mathbb{V}$ two subsets of $\mathbb{Z}$, we denote $\mathbb{U} \leq \mathbb{V}$ when for all $i \in \mathbb{U}$ and $j \in \mathbb{V}$, then $i \leq j$.

Definition 5.3. *We say that a subset $\mathbb{U}$ of $\mathbb{Z}$ is connected when for all $i \leq j \leq k$ in $\mathbb{U}$,*

$$i, k \in \mathbb{U} \Rightarrow j \in \mathbb{U}.$$

Moreover, for two subsets $\mathbb{U}, \mathbb{V}$, we denote $\gamma(\mathbb{U}, \mathbb{V})$ the smallest connected subset of $\mathbb{Z}$ containing both $\mathbb{U}$ and $\mathbb{V}$.

Proposition 5.4. *Let X be some $\mathbb{Z}$-subshift and f some integer function. The subshift X is f-block gluing if and only if it is f-irreducible.*

Proof. 1. **Direct implication:**

It is clear that the f-irreducibility property implies the f-block gluing one. We shall prove the converse implication. First, let us prove some property of pairs of subsets of $\mathbb{Z}$.

2. **Inclusion of two finite sets into an alternation of connected sets:**

Let $\mathbb{U}, \mathbb{V}$ be two non-empty subsets of $\mathbb{Z}$ such that $\delta(\mathbb{U}, \mathbb{V}) > 0$. Then there exist $\mathbb{U}', \mathbb{V}'$ such that:

(a) $\mathbb{U} \subset \mathbb{U}'$ and $\mathbb{V} \subset \mathbb{V}'$,

(b) $\delta(\mathbb{U}', \mathbb{V}') = \delta(\mathbb{U}, \mathbb{V})$,

(c) There exists two sequences of connected non-empty subsets of $\mathbb{Z}$ $(\mathbb{U}'_i)_{i=1\ldots k}$ and $(\mathbb{V}'_i)_{i=1\ldots l}$, with $l \in \{k-1, k, k+1\}$ such that $\mathbb{U}' = \bigcup_i \mathbb{U}'_i$, $\mathbb{V}' = \bigcup_i \mathbb{V}'_i$ and one of the following statements is true:

- $\mathbb{U}'_1 \leq \mathbb{V}'_1 \leq \mathbb{U}'_2 \leq .... \leq \mathbb{V}'_l \leq \mathbb{U}'_k$
- $\mathbb{U}'_1 \leq \mathbb{V}'_1 \leq \mathbb{U}'_2 \leq .... \leq \mathbb{U}'_k \leq \mathbb{V}'_l$
- $\mathbb{V}'_1 \leq \mathbb{U}'_1 \leq \mathbb{V}'_2 \leq .... \leq \mathbb{V}'_l \leq \mathbb{U}'_k$
- $\mathbb{V}'_1 \leq \mathbb{U}'_1 \leq \mathbb{V}'_2 \leq .... \leq \mathbb{U}'_k \leq \mathbb{V}'_l$.

Indeed, there exist two sequences $(\mathbb{U}_i)_i$ and $(\mathbb{V}_i)_i$ of minimal sets verifying that

- $\mathbb{U} = \bigcup_i \mathbb{U}_i$, $\mathbb{V} = \bigcup_i \mathbb{V}_i$,
- for all i $\mathbb{U}_i$ and $\mathbb{V}_i$ are connected,
- all the sets $\mathbb{U}_i$ and $\mathbb{V}_i$ are disjoint.

In order to obtain $\mathbb{U}'$, one execute the following operations: for i, k such that $\mathbb{U}_i \leq \mathbb{U}_k$ and there exists no j such that $\mathbb{U}_i \leq \mathbb{V}_j \leq \mathbb{U}_k$, replace $\mathbb{U}_i \bigcup \mathbb{U}_k$ in the definition of $\mathbb{U}$ by $\gamma(\mathbb{U}_i, \mathbb{U}_k)$. We obtain $\mathbb{V}'$ by similar operations.

3. **Reduction to the gluing on alternation of connected sets:**

   Let X be some f-block gluing $\mathbb{Z}$-subshift and f some integer function. Consider $\mathbb{U}, \mathbb{V}$ finite subsets of $\mathbb{Z}$ such that
   $$\delta(\mathbb{U}, \mathbb{V}) \geq f(\max(\delta(\mathbb{U}), \delta(\mathbb{V}))),$$
   and u, v patterns on U, V that are globally admissible for Σ. We complete u and v into respectively u' and v' patterns on $\mathbb{U}', \mathbb{V}'$ obtained as above from $\mathbb{U}$ and $\mathbb{V}$. This is sufficient to prove that there exists some configuration x of Σ such that $x_{\mathbb{U}'} = u'$ and $x_{\mathbb{V}'} = v'$. Without loss of generality, we assume that $\mathbb{U}'_1 \leq \mathbb{V}'_1 \leq \mathbb{U}'_2 \leq .... \leq \mathbb{V}'_l \leq \mathbb{U}'_k$.

4. **Successive application of the block gluing property on the restrictions to connected sets:**

   Let us denote $u^1, ...u^k$ the restriction of u' on $\mathbb{U}'_1, ..., \mathbb{U}'_k$, and $v^1, ..., v^l$ the restriction of v' on $\mathbb{V}'_1, ..., \mathbb{V}'_l$. Let us denote $n = \max(\delta(\mathbb{U}), \delta(\mathbb{V}))$. Then $\delta(\mathbb{U}'_1, \mathbb{V}'_1) \geq f(n)$. We have the inequality $n \geq \max(\delta(\mathbb{U}'_1), \delta(\mathbb{V}'_1))$: this comes from the fact that these sets are contained respectively in $\mathbb{U}'$ and $\mathbb{V}'$. As a consequence, from the block gluing property, there exists some globally admissible pattern w on $\gamma(\mathbb{U}'_1, \mathbb{V}'_1)$ whose restriction on $\mathbb{U}'_1$ (resp. $\mathbb{V}'_1$) is u^1 (resp. v^1). Since n is still greater than $\delta(\gamma(\mathbb{U}'_1, \mathbb{V}'_1))$ and $\delta(\mathbb{U}'_2)$, and that these two sets are spaced by $f(n)$, there exists some globally admissible pattern on $\gamma(\gamma(\mathbb{U}'_1, \mathbb{V}'_1), \mathbb{U}'_2)$ whose restriction on $\gamma(\mathbb{U}'_1, \mathbb{V}'_1)$ is w and on $\mathbb{U}'_2$ is u^2. Repeating this kind of argument, we get that there exists some globally admissible pattern on $\gamma(\mathbb{U}', \mathbb{V}')$ whose restriction on $\mathbb{U}'$ is u' and on $\mathbb{V}'$ is v'.

$\square$

5.2.3 Decidability

A language $\mathcal{L}$ on $\mathbb{Z}^d$ and on alphabet $\mathcal{A}$ is said to be **decidable** when there exists an algorithm which, given as input a finite pattern u, halts and returns 1 if u is in $\mathcal{L}$ and 0 otherwise.

As well, a $\mathbb{Z}^d$-subshift is **decidable** when its language is decidable.

5.2.4 Irreducibility and decidability for SFT

It is a well-known fact that multidimensional SFT ($d \geq 2$) are not decidable in general. This is sometimes referred to as the undecidability of the extension problem. However, subshifts of finite type with some dynamical property (for instance block gluing property, or minimality) are known to be decidable. In this section we prove that f-irreducible SFT for a function such that $f(n) = o(n)$ are decidable. The arguments of the proof are similar as in the proof of Corollary 3.5 in [HM10]. This corollary implies that $O(1)$-irreducible subshifts are decidable.

Let X be a d-dimensional SFT for some $d \geq 1$, defined by a finite set $\mathcal{F}$ of forbidden r-blocks for some $r \geq 1$. In particular the diameter of the support of these blocks is equal to r.

In the following, we denote and $\mathbb{D}_k = [\![-k-r, k+r]\!]^d \backslash [\![-k, k]\!]^d$ for all k integer.

Lemma 5.5. *Assume that X is f-irreducible for some function f such that $f(k) = o(k)$. Let n be some integer and u a pattern on $[\![-n,n]\!]^d$. This pattern is globally admissible for X if and only if there exist some integers N_0 and m such that*

- *$m \geq n$ and $N_0 \geq m$,*
- *for all $N \geq N_0$ and for all v locally admissible pattern on $[\![-N,N]\!]^d$, then there exists a pattern p on $[\![-N,N]\!]^d$ which is locally admissible such that $p_{\mathbb{D}_m} = v$ and $p_{[\![-n,n]\!]^d} = u$.*

Proof. 1. **Indirect implication:**

If u verifies this last condition, by compactness, it is globally admissible.

2. **Direct implication:**

Assume that the pattern u is globally admissible.

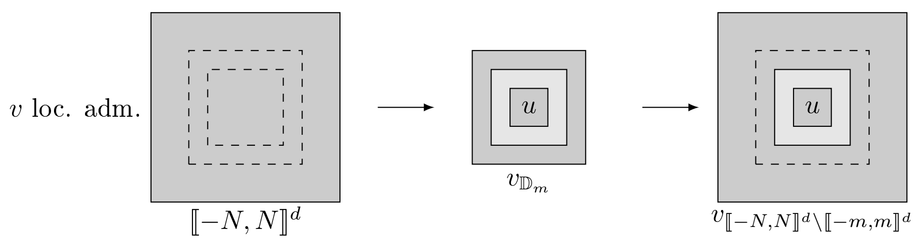
- **Asymptotic property of f and application of the f-irreducibility:**
For all $k \geq n$, $[\![-n,n]\!]^d$ and $\mathbb{D}_k$ have both diameter at most $2k+2r$ and

$$\delta([\![-n,n]\!]^d, \mathbb{D}_k) = k - n.$$

Let us denote m the minimal integer such that

$$m - n \geq f(2m+2r),$$

which exists since $f(k) = o(k)$. From the f-irreducibility property of the subshift X, we have that for any globally admissible v patterns on $\mathbb{D}_m$ and $[\![-n,n]\!]^d$, there exists a configuration x of X such that $x_{\mathbb{D}_m} = v$ and $x_{[\![-n,n]\!]^d} = u$.



- **Use of Lemma 2.11 on the relation between globally admissible and locally admissible patterns:**
For any non globally admissible pattern w on $\mathbb{D}_m$, there exists some N_w such that for all $N \geq N_w$, no locally admissible pattern on $[\![-N,N]\!]^d$ has its restriction on $\mathbb{D}_m$ equal to w. Let us denote N_0 the maximum of the numbers N_w, for such a pattern w (there are finitely many of them).
For all $N \geq N_0$, any locally admissible patterns v on $[\![-N,N]\!]^d$ has globally admissible restriction on $\mathbb{D}_m$.
- **Combining the two last points:**
Consider some integer $N \geq N_0$, and any locally admissible pattern v on $[\![-N,N]\!]^d$. Using the first point, there exists some locally admissible pattern on the set $[\![-m-r,m+r]\!]^d$ whose restriction on $[\![-n,n]\!]^d$ is u and restriction on $\mathbb{D}_m$ is $v_{\mathbb{D}_m}$. Then completing this pattern on $[\![-N,N]\!]^d$ with the restriction of v on

$$[\![-N,N]\!]^d \setminus [\![-m-r,m+r]\!]^d$$

does not change the local admissibility of this pattern. See Figure 2.

□

Theorem 5.6. *Let X a $\mathbb{Z}^d$-SFT on some alphabet $\mathcal{A}$. If it is f-irreducible with f a computable function such that $f(n) = o(n)$, then it is decidable.*

Proof. Let u be some d-dimensional finite pattern on alphabet $\mathcal{A}$. In order to decide if u is in the language of X or not, one can enumerate successively for all n the locally admissible patterns having size n and test the conditions of Lemma 2.11 and Lemma 5.5. If the conditions of Lemma 5.5 are verified, then output 1 (meaning that the pattern is globally admissible). If the conditions of Lemma 2.11 are verified, output 0 (this means that the pattern is not globally admissible). If they are not true Since u is either globally admissible or not globally admissible, there is some n for which one of the conditions is verified. Hence this sequence of instructions ends. This means that X is decidable. □

This result motivates our study: in the following, we extend the class of $o(n)$-irreducible SFT (which includes $O(1)$-irreducible ones) to the class of $o(n)$-irreducible decidable subshifts, which is intuitively much larger than its restriction to SFT.

5.3 Computability of the entropy of irreducible subshifts

In this section we prove Theorem 1.2.

Here is the organization of the section:

- In Section 5.3.1 we prove that when f verifies the summability condition, then the entropy of f-irreducible effective subshifts is computable.
- We prove in Section 5.3.2 that when this condition is not verified, then all the Π_1-computable numbers are the entropy of an f-irreducible subshift.

In the following of this chapter, we will use the following definition:

Definition 5.7. *Let $d \geq 1$, and $\mathcal{C}$ some class of d-dimensional decidable subshifts on alphabet $\mathcal{A}$. We say that there is an algorithm that computes the entropy of the subshifts in the class $\mathcal{C}$ when there is an algorithm that:*

- *on two integers n, m input, where the first one represents a Turing machine deciding the language of a d-dimensional decidable subshift in $\mathcal{C}$ and some integer m, the algorithm halts.*
- *It outputs some rational number $r_{n,m}$*

$$|h(X) - r_{n,m}| \leq 2^{-m},$$

where X is the decidable subshift decided by the nth Turing machine.

Remark 36. *In Definition 5.7, the same assertion with an algorithm taking as input an integer n representing any Turing machine would have no meaning. That is why we restrict the possible inputs. The intuitive idea is that there is a uniform way to compute the entropy of subshifts in this class.*

5.3.1 Under the threshold

In this section, we prove Theorem 5.8 which corresponds to the case when f is under the threshold. We first prove this for $d = 1$ and then extend the proof to $d \geq 1$.

Theorem 5.8. *Let f be a non-decreasing computable function such that the series having general term $\frac{f(n)}{n^2}$ converges at a computable rate. Then there is an algorithm that computes the entropy of f-block gluing (and as a consequence f-irreducible ones) one-dimensional decidable subshifts.*

Remark 37. *We also have the following: there exists an algorithm which, given as input some positive real number $c > 0$ and m an integer representing a Turing machine that decides the language of some $\mathbb{Z}^d$-subshift which is c-block gluing, computes the entropy of this subshift. We don't write the proof of this statement, since it can be easily derived from the proof of Theorem 5.8.*

Proof. Let us describe how to find approximation of the entropy of a one dimensional f-block gluing decidable subshift X having alphabet $\mathcal{A}$.

- **From block gluing to an inequality on the complexity function:**

  By definition of the f-block gluing property, for any two words $u, v \in \mathcal{L}_n(X)$, there exists a word $w \in \mathcal{A}^{f(n)}$ such that $uwv \in \mathcal{L}_{2n+f(n)}(X)$. Because u, v can be chosen freely we have

$$|\mathcal{L}_n(X)|^2 \leq |\mathcal{L}_{2n+f(n)}(X)| \leq |\mathcal{A}|^{f(n)} \cdot |\mathcal{L}_{2n}(X)|,$$

  the second inequality coming from a decomposition of the $2n + f(n)$ length words into a length $2n$ word and some $f(n)$ length word. In particular

$$|\mathcal{L}_n(X)|^2 \leq |\mathcal{A}|^{f(n)} \cdot |\mathcal{L}_{2n}(X)|.$$

  By taking the logarithm of this inequality and dividing by $2n$, we get:

$$\frac{\log_2 |\mathcal{L}_n(X)|}{n} \leq \frac{\log_2(|\mathcal{L}_{2n}(X)|)}{2n} + \log_2(|\mathcal{A}|)\frac{f(n)}{2n}$$

- **Iteration of the inequality:**

  The previous inequality being true from any n, we apply it for the integers of the sequence $(2^k n)_{k \geq 0}$. Using consecutively the $l \geq 1$ first inequalities we get that

$$\frac{\log_2(|\mathcal{L}_n(X)|)}{n} \leq \frac{\log_2(|\mathcal{L}_{2^l n}(X)|)}{2^l n} + \log_2(|\mathcal{A}|) \sum_{k=1}^{l} \frac{f(2^{k-1}n)}{2^k n}$$

- **Taking the limit of the inequality using the convergence of the series:**

  Using the fact that $\frac{\log_2(|\mathcal{L}_n(X)|)}{n}$ converges towards $h(X)$ and that the series of the $f(2^k)/2^k$ converges, for all $n \geq 1$ we have:

$$\frac{\log_2(|\mathcal{L}_n(X)|)}{n} \leq h(X) + \log_2(|\mathcal{A}|) \sum_{k=1}^{\infty} \frac{f(2^{k-1}n)}{2^k n}$$

  We apply this inequality on 2^n and get

$$\frac{\log_2(|\mathcal{L}_{2^n}(X)|)}{2^n} \leq h(X) + \frac{\log_2(|\mathcal{A}|)}{2} \sum_{k=0}^{\infty} \frac{f(2^{k+n})}{2^{k+n}}.$$

  This means that for all n,

$$h(X) \leq \frac{\log_2(|\mathcal{L}_{2^n}(X)|)}{2^n} \leq h(X) + \frac{\log_2(|\mathcal{A}|)}{2} \sum_{k=n}^{\infty} \frac{f(2^k)}{2^k}.$$

Since f is non-decreasing, we have that for all k,

$$\frac{f(2^k)}{2^k} \leq \frac{1}{2^k} \sum_{i=2^k}^{2^{k+1}-1} \frac{f(i)}{i} \leq \frac{1}{2} \sum_{i=2^k}^{2^{k+1}-1} \frac{f(i)}{i^2}$$

The first inequality means that the mean of the values of the function $n \mapsto f(n)/n$ between 2^k and $2^{k+1}-1$ is greater than its value on 2^k. For the second inequality we use that on this interval $1/i \geq 1/2^{k+1}$.

- **Approximations of the entropy using the computability of the convergence rate:**

  Using Remark 31, the series $\sum_{k \geq n} \frac{f(2^k)}{2^{k+1}}$ converges at a computable rate. This means that there exists a computable function $t \mapsto n(t)$ such that for all t,

  $$\sum_{k=n(t)+1}^{\infty} \frac{f(2^k)}{2^k} \leq \frac{2^{-t-1}}{\log_2(|\mathcal{A}|)}$$

  This implies that
  $$\left| h(X) - \frac{\log_2(|\mathcal{L}_{2^{n(t)}}|)(\Sigma)}{2^{n(t)}} \right| \leq 2^{-t-1}.$$

An algorithm computing the entropy of the subshift X_m decided by the mth Turing machine up to a precision 2^{-t} runs as follows:

1. It computes $n(t)$.
2. It computes the cardinality of $\mathcal{L}_{2^{n(t)}}(X_m)$. This is possible by simulating the mth Turing machine with a universal machine.
3. It computes a rational approximation of $\frac{\log(|\mathcal{L}_{2^{n(t)}}|(X_m))}{2^n(t)}$ up to a precision 2^{-t-1}. This rational number is thus a rational approximation of $h(\Sigma)$ up to precision 2^{-t}. This is possible using the computability of the logarithm function.

We conclude that the entropy of f-irreducible decidable subshifts is uniformly computable. □

We now adapt this proof to the case of d-dimensional subshifts, for $d \geq 2$.

Theorem 5.9. *Let $d \geq 1$ and f be a non-decreasing computable function such that the series having general term $\frac{f(n)}{n^2}$ converges with computable rate. Then there is an algorithm that computes the entropy of f-block gluing d-dimensional decidable subshifts.*

Proof. Let $d \geq 1$, and X a d-dimensional decidable subshift which is f-block gluing. Denote for $n_1, \ldots, n_k \in \mathbb{N}$,
$$\mathbb{C}[n_1, \ldots, n_d] = [\![0, n_1 - 1]\!] \times ... \times [\![0, n_d - 1]\!].$$
The diameter of this set is $N = \max_k n_k$.

- **Application of the f block gluing property:**

  From the f-block gluing property of X applied on two patterns u, v having support $\mathbb{C}[n_1, \ldots, n_d]$, there exists some pattern p on $\mathbb{C}[n_1, n_2 ... n_{k-1}, 2n_k + f(N), n_{k+1}, ... n_d]$ whose restriction on $\mathbb{C}[n_1, \ldots, n_d]$ is u and on $(n_k + f(N))\mathbf{e}^k + \mathbb{C}[n_1, \ldots, n_d]$ is v.

Considering the case $k=1$, and since u and v can be chosen freely,

$$\begin{array}{rcl} |\mathcal{L}_{\mathbb{C}[n_1,\dots,n_d]}|^2 & \leq & |\mathcal{L}_{\mathbb{C}[2n_1+f(N),n_2,\dots n_d]}| \\ & \leq & |\mathcal{L}_{\mathbb{C}[2n_1+f(N),n_2,\dots,n_d]}| \cdot |\mathcal{A}|^{n_2\times\dots\times n_d \times f(N)} \\ & \leq & |\mathcal{L}_{\mathbb{C}[2n_1,n_2,\dots,n_d]}| \cdot |\mathcal{A}|^{N^{d-1}f(N)} \end{array}$$

The second inequality comes from a decomposition of the pattern p into a pattern on $\mathbb{C}[2n_1, n_2, .., n_d]$ and a pattern on $\mathbb{C}[f(N), n_2, .., n_d]$.

- **Iterating the inequality on the dimension parameter:**

  The diameter of $\mathbb{C}[2n_1, n_2, ..., n_d]$ is less than $2N$, so we can apply the same argument to this set and get:

$$|\mathcal{L}_{\mathbb{C}[2n_1,n_2,\dots,n_d]}(X)|^2 \leq |\mathcal{L}_{\mathbb{C}[2n_1,2n_2,n_3,\dots,n_d]}(X)| \cdot |\mathcal{A}|^{2N^{d-1}f(2N)}.$$

  Because the diameter does not change (it is still $2N$), we can re-apply this on the set $C[2n_1, 2n_2, n_3, \dots, n_d]$ and get

$$|\mathcal{L}_{\mathbb{C}[2n_1,2n_2,n_3,\dots,n_d]}(X)|^2 \leq |\mathcal{L}_{\mathbb{C}[2n_1,2n_2,n_3,\dots,n_d]}(X) \cdot |\mathcal{A}|^{2N^{d-1}f(2N)},$$

  etc, until we get

$$|\mathcal{L}_{C[2n_1,\dots,2n_{d-1},n_d]}(X)|^2 \leq |\mathcal{L}_{C[2n_1,\dots,2n_d]}(X)| \cdot |\mathcal{A}|^{2N^{d-1}f(2N)}.$$

  Applying successively these inequalities, we get

$$|\mathcal{L}_{\mathbb{C}[n_1,\dots,n_d]}(X)^{2^d}| \leq |\mathcal{L}_{\mathbb{C}[2n_1,\dots,2n_d]}(X)| \cdot (|\mathcal{A}|^{(2N)^{d-1}f(2N)})^{d-1} \cdot |\mathcal{A}|^{N^{d-1}f(N)}$$

  Because f is non decreasing, we have

$$|\mathcal{L}_{\mathbb{C}[n_1,\dots,n_d]}(X)|^{2^d} \leq (|\mathcal{A}|^{(2N)^{d-1}f(2N)})^d.$$

- **For a block:**

  This implies, taking $n_1 = ... = n_d = n$ for some n, that

$$|\mathcal{L}_{\mathbb{C}[n,\dots,n]}(X)|^{2^d} \leq |\mathcal{L}_{\mathbb{C}[2n,\dots,2n]}(X)| \cdot (|\mathcal{A}|^{(2n)^{d-1}f(2n)})^d.$$

  Taking the logarithm and then dividing by $(2n)^d$:

$$\frac{\log_2(|\mathcal{L}_{\mathbb{C}[n,\dots,n]}(X)|)}{n^d} \leq \frac{\log_2(|\mathcal{L}_{\mathbb{C}[2n,\dots,2n]}(X)|)}{(2n)^d} + \log_2(|\mathcal{A}|) d \frac{f(2n)}{2n}.$$

The end of the proof is similar to the proof of Theorem 5.8.

□

Corollary 5.10. *For a function f that verifies the same conditions as Theorem 5.8, there is an algorithm which computes the entropy of f-irreducible d-dimensional decidable subshifts.*

Example 5.11. *For any $1 > \varepsilon > 0$, there exists an algorithm that computes the entropy of $n/(\log(n+2)^{1+\varepsilon}$-block gluing decidable subshifts, for ε sufficiently small.*

Proof. Let $\varepsilon > 0$ and $f : n \mapsto n/(\log(n+2))^{1+\varepsilon}$. Thanks to Remark 34, any f-irreducible subshift is g-irreducible, with $g : n \mapsto n/(\log(n)+2)^{1+r}$, with a rational number r such that $0 < r < \varepsilon$. In order to compute the entropy of f-irreducible decidable subshifts, we consider these subshifts as g-irreducible ones.

What we have left to prove is that the series

$$\sum_k \frac{g(k)}{k^2} = \sum_k \frac{1}{k\log(k)^{1+r}}$$

converges at a computable convergence rate. We use the following inequality, for all n,

$$\sum_{k\geq n} \frac{1}{k\log(k)^{1+r}} \leq \int_{n-1}^{\infty} \frac{dt}{t\log(t)^{1+r}} = \frac{1}{r\log(n-1)^r},$$

using the fact that $t \mapsto 1/t\log(t)^{1+r}$ is non-decreasing.

Thus, the series converges with rate

$$n : t \mapsto \left\lceil \exp\left(\left\lceil \frac{2^t}{\varepsilon}\right\rceil^{1/\varepsilon}\right) + 1 \right\rceil,$$

which is a computable function. Indeed, with this function

$$\frac{1}{r\log(n-1)^r} \leq 2^{-t}$$

for all t, and thus

$$\sum_{k\geq n(t)} \frac{g(k)}{k^2} \leq 2^{-t}.$$

□

5.3.2 Above the threshold

In this section, we consider decidable subshifts whose irreducibility rate are above the threshold, and prove the following theorem:

Theorem 5.12. *Let $\alpha > 0$ be a Π_1-computable real number and $f : \mathbb{N} \to \mathbb{N}$ be a computable non-decreasing function such that*

$$\sum_{n=1}^{\infty} \frac{f(n)}{n^2} = +\infty.$$

Then there exists a decidable f-irreducible one dimensional subshift with entropy α.

This theorem will be proved first in this section for $d = 1$ using **bounded density subshifts**, introduced by B. Stanley [Sta13] (as bounded density *shifts*), presented in Section 5.3.2.1. The proof for this case is given by a construction presented in Section 5.3.2.10 and uses the sections before this one, in which we prove properties of bounded density subshifts. Then this result is extended to $d \geq 1$ in Section 5.3.2.11.

In the following, f is a given function which verifies the conditions of Theorem 5.12. In all the following, we denote $F : n \mapsto 2n + f(n)$.

5.3.2.1 Bounded density subshifts

Definition 5.13 (Bounded Density subshift)**.** *Let $p = (p_n)_{n\geq 1}$ be a sequence of positive integers. Let us denote $X_p \subset \{0,1\}^{\mathbb{Z}}$ the one dimensional subshift defined by the set of forbidden words $\mathcal{F}^p = \bigcup_{n\geq 1} \mathcal{F}_n^p$, where for all n*

$$\mathcal{F}_n^p = \{u \in \mathcal{A}^n \ : \ \#_1 u > p_n\}.$$

Example 5.14. 1. *The bounded density subshift defined by the sequence $(p_n)_n = (1,1,2,3,4,...)$ is the Golden mean shift defined on alphabet $\{0,1\}$ by the set of forbidden words $\mathcal{F} = \{11\}$. Indeed, since a length 2 word in the language of X_p can not contain more than one symbol 1, the word 11 is forbidden. Hence this subshift is contained in the Golden mean shift. Reciprocally, let x be a configuration of the Golden mean shift. For any word u appearing in x and having length $n \geq 2$, u contains at least one symbol 0 (if not then 11 appears in this word). The number of symbols 1 in this word is thus less than $n-1$. As a consequence, x verifies all the conditions to be in X_p.*

   Equivalently, this subshift can be defined by the sequence $p' = (1,1,2,2,3,3,4,4,...)$. Indeed, because $p' \leq p$, we have $X_{p'} \subset X_p$. Moreover, if $x \in X_p$, we have to show that if a length $n \geq 2$ word appears in x, then it contains at most $\lceil n/2 \rceil$ symbols 1. Let u be such a word. For each 1 it contains, there is a symbol 0 on the right in the word u, with one exception if the last symbol of u is 1. Hence $2\#_a u - 1 \leq n$, and $\#_a u \leq (n+1)/2 \leq \lceil n/2 \rceil$.

2. *For $n \in \mathbb{N}$, define X_n the bounded density subshift associated to the sequence p^n defined as follows:*

   - *if the n-th Turing machine starting on an empty input stops before the kth step, then $p_k^n = p_{k-1}$;*
   - *otherwise, $p_k^n = p_{k-1} + 1$.*

   If the n-th Turing machine never stops on the empty input, then $p_n = n$ for all n and $X_n = \{0,1\}^{\mathbb{Z}}$ the full shift, and has entropy 1. On the other hand, if the n-th Turing machine stops at some point, then p^n is ultimately constant and X_n has entropy zero. Indeed, this derives directly from Lemma 5.17.

5.3.2.2 Bounded density subshifts having computable density function are decidable

Since we use bounded density subshifts in the proof of Theorem 5.12, we need to prove that they are decidable (Lemma 5.16). The first lemma (Lemma 5.15) is used in the proof of the second one (Lemma 5.16), and is re-used afterwards.

Lemma 5.15. *Let p be some non-decreasing sequence of integers, and w a word locally admissible for X_p (where the set of forbidden patterns corresponds to the one used in Definition 5.13 to define this subshift). Then for all m, n integers, $0^m w 0^n$ is globally admissible in X_p.*

Proof. The element x of $\{0,1\}^{\mathbb{Z}}$ such that

- $x_{[\![0,|w|-1]\!]} = w$
- and for all $\mathbf{i} \in \mathbb{Z} \backslash [\![0,|w|-1]\!]$, $x_{\mathbf{i}} = 0$.

is an element of X_p. Indeed, any subword of x is amongst the following types:

1. $0^k u$, where $k \geq 0$ and u some prefix of w,
2. $v 0^k$, where $k \geq 0$, and v is some suffix of w

3. $0^m w 0^n$, where $n, m \geq 0$.

In the first case, the number of 1 symbols in the word is smaller than $p_{|u|}$ since w is locally admissible. Since p is non-decreasing, $p_{|u|+k} \geq p_{|u|}$. The two other cases are similar.

As a consequence, for all $n, m \geq 0$, $0^m w 0^n$ is globally admissible. □

Lemma 5.16. *If p is non-decreasing and computable, then X_p is decidable.*

Proof. Since p is non-decreasing, Lemma 5.15 implies that any locally admissible pattern of Σ_p is globally admissible. Hence an algorithm that decides if an input word $w \in \{0,1\}^l$ is in $\mathcal{L}(\Sigma_p)$ runs as follows:

- it computes the value of p_k for all $k \leq l$;
- then computes the set of forbidden patterns having length $\leq l$;
- check if one of these patterns appears in w.
- if this is the case, return 0 ; else, return 1. □

The output 0 signifies that the word is in the language, and 1 signifies it is not.

5.3.2.3 Zero limit density implies zero entropy

In this section, we prove a technical lemma related to the entropy of bounded density subshifts. This lemma will be used in the construction for the proof of Theorem 5.12 in order to ensure that the aimed number is indeed the entropy of the constructed subshift.

For p a sequence of positive integers, we call limit density the number $\lim \frac{p_n}{n}$ when it exists.

Lemma 5.17. *Let p be some sequence such that $\frac{p_n}{n} \to 0$. Then the entropy of X_p is zero.*

In other words, if the limit density is zero, then the entropy is also zero.

As well, if $\inf_n \frac{p_n}{n} = 0$, then the entropy of X_p is zero.

Proof. Consider such a sequence p and some integer n_0 such that for $n \geq n_0$,

$$p_n \leq \left\lfloor \frac{n}{2} \right\rfloor.$$

Since for all n, any length n word in the language of X_p has less than p_n symbols equal to 1, the number of these words is smaller than

$$\sum_{k=0}^{p_n} \binom{n}{k},$$

which is smaller than

$$(p_n + 1)\binom{n}{p_n},$$

since $k \mapsto \binom{n}{k}$ is non-increasing on $\left[\!\left[0, \left\lfloor \frac{n}{2} \right\rfloor\right]\!\right]$.

The entropy of X_p is thus smaller than

$$\frac{\log_2(p_n + 1)}{n} + \frac{1}{n} \log_2 \left(\binom{n}{p_n} \right).$$

Since for all n,

$$\binom{n}{p_n}\left(\frac{p_n}{n}\right)^{p_n}\left(1-\frac{p_n}{n}\right)^{n-p_n} \leq \sum_{k=0}^{n}\binom{n}{k}\left(\frac{k}{n}\right)^k\left(1-\frac{k}{n}\right)^{n-k} = 1,$$

applying the logarithm function to this inequality leads to

$$h(\Sigma_p) \leq \frac{1}{n}\log_2\left(\binom{n}{p_n}\right) \leq -\frac{p_n}{n}\log_2\left(\frac{p_n}{n}\right) - \left(1-\frac{p_n}{n}\right)\log_2\left(1-\frac{p_n}{n}\right),$$

Then, if p_n/n tends towards 0, or if $\inf_n \frac{p_n}{n} = 0$, the right-hand side of this inequality tends towards 0 and $h(X_p) = 0$. This is a standard argument, and the right-hand side of the previous inequality is often called binary entropy. □

5.3.2.4 Piecewise linear density functions

In the proof of Theorem 5.12, the bounded density subshifts that we use are more precisely given by a "piecewise linear" density sequence p, as defined in Definition 5.18.

Definition 5.18 (Piecewise linear density functions)**.** *Let $\beta = (\beta_k)_{k\geq 0}$ be a sequence of positive rational numbers and $f : \mathbb{N} \to \mathbb{N}$ a computable function. Recall that F denotes the function $n \mapsto 2n + f(n)$. We define the function $\varphi(f,\beta) : \mathbb{R}_+ \to \mathbb{R}_+$ such that for all $n \geq 1$ and $t \in [0, F^{n+1}(1) - F^n(1)]$,*

$$\varphi(f,\beta)(F^n(1)+t) = \varphi(f,\beta)(F^n(1)) + t.\beta_n.$$

Then, we define $p(f,\beta) : \mathbb{N} \to \mathbb{N}$ by

$$p(f,\beta)_n = \left\lceil \varphi(f,\beta)(n) \right\rceil.$$

If β is a finite sequence, then we identify this sequence with the infinite sequence β' defined as follows:

$$\beta'_k = \begin{cases} \beta_k & \text{if } k \leq n \\ \beta_n & \text{if } k > n \end{cases}$$

By abuse of notation, we denote $\varphi(f,\beta) = \varphi(f,\beta')$ and $p(f,\beta) = p(f,\beta')$. The bounded density subshift obtained with the sequence $p(f,\beta)$, where β is a sequence, finite or infinite, is denoted $\Sigma_{f,\beta}$.

Figure 5.1 illustrates schematically this definition.

Lemma 5.19. *Let β be a non-increasing sequence of positive rational numbers such that $\beta_n \to 0$. Then the entropy of $X_{f,\beta}$ is zero.*

Proof. Since β is non-increasing, for all $n \geq 1$ and $t \geq 0$,

$$\varphi(f,\beta)(F^n(1)+t) \leq \varphi(f,\beta)(F^n(1)) + t.\beta_n,$$

and as a consequence,

$$p(f,\beta)_{F^n(1)+t} \leq \varphi(f,\beta)(F^n(1)) + t.\beta_n + 1,$$

Hence

$$\frac{p(f,\beta)_{F^n(1)+t}}{F^n(1)+t} \leq \frac{\varphi(f,\beta)(F^n(1)) + t.\beta_n + 1}{F^n(1)+t} \to \beta_n$$

when $t \to +\infty$. This means that

$$\limsup_k \frac{p_k}{k} \leq \beta_n.$$

Since this is true for all n and that $\beta_n \to 0$, then $\frac{p_n}{n} \to 0$. From Lemma 5.17, we get that the entropy of $X_{f,\beta}$ is zero. □

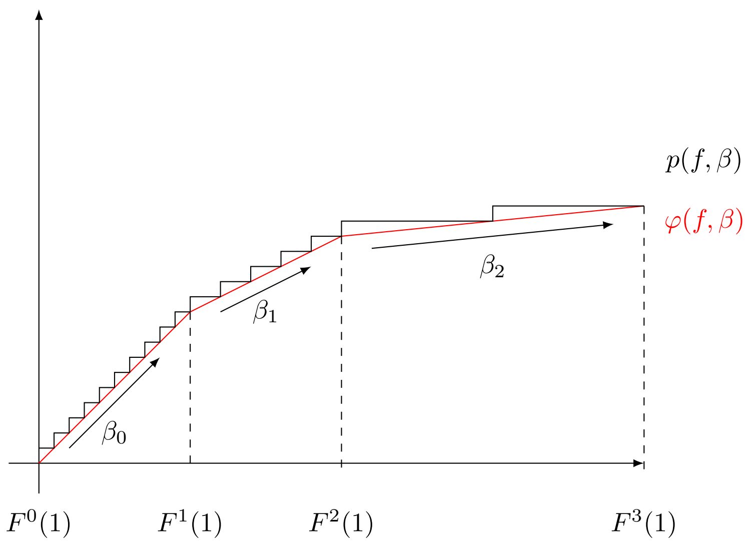


Figure 5.1: Illustration of Definition 5.18.

Lemma 5.20. *Let β be a computable sequence of rational numbers. Then the subshift $X_{f,\beta}$ is decidable.*

Proof. This is an application of Lemma 5.16. The conditions of this lemma are verified if β is computable, since then $p(f,\beta)$ is computable, and always non-decreasing by definition. □

5.3.2.5 Concaveness and irreducibility

When the density function is furthermore concave, we prove that a condition on particular values of the function implies the f-irreducibility of the subshift. In the construction for the proof of Theorem 5.12, it will be sufficient to ensure this condition so that the obtained subshift is f-irreducible.

Definition 5.21. *A sequence $p : \mathbb{N} \to \mathbb{N}$ is said to be **concave** when there exists a concave function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ such that for all $n \geq 1$,*

$$p_n = \left\lceil \varphi(n) \right\rceil.$$

Lemma 5.22. *Let p be a concave sequence of integers. For all integers $k, l \geq 0$ and $n \geq 1$,*

$$p_{n+k+l} - p_{n+l} \leq p_{n+k} - p_n + 2.$$

Proof. Let $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ be such that for all $n \geq 1$,

$$p_n = \left\lceil \varphi(n) \right\rceil.$$

Then for all $k, l \geq 0$ and $n \geq 1$,

$$\begin{array}{rcl} p_{n+k+l} + p_n - p_{n+l} - p_{n+k} & = & \left\lceil \varphi(n+k+l) \right\rceil + \left\lceil \varphi(n) \right\rceil - \left\lceil \varphi(n+l) \right\rceil - \left\lceil \varphi(n+k) \right\rceil \\ & \leq & \varphi(n+k+l) + \varphi(n) - \varphi(n+k) - \varphi(n+l) + 2 \\ & \leq & 2 \end{array},$$

since the function φ is concave. □

Lemma 5.23. *Assume that p is a concave sequence and that for all $n \geq 1$, $p_{F(n)} \geq 2p_n + 4$. Then X_p is f-irreducible.*

Proof. Let p be some sequence which verifies these conditions.

Consider some integer $n \geq 1$ and two words u, v in $\mathcal{L}(X_p)$ with $|u| \leq n$, $|v| \leq n$. We prove that the word $u0^{f(n)}v$ is in the language of X_p. From Lemma 5.15, it is sufficient to prove that this word is locally admissible.

Let w be a subword of $u0^{f(n)}v$. It is amongst the following types:

- $u'0^k$, where u' is a suffix of u and $k \leq f(n)$,
- $0^k v'$, where v' is a prefix of v and $k \leq f(n)$,
- $u'0^{f(n)}v'$, where u' is a suffix of u and v' is a prefix of v.

In the two first cases, this word is in $\mathcal{L}(X_p)$, using Lemma 5.15.

In the third case, since u' and v' are in $\mathcal{L}(X_p)$, we have $\#_1 u' \leq p_{|u'|}$ and $\#_1 v' \leq p_{|v'|}$. Therefore $\#_1 w \leq p_{|u'|} + p_{|v'|}$.

Since p is a concave sequence, as a consequence of Lemma 5.22, we have:

- $p_n - p_{|u'|} + 2 \geq p_{2n+f(n)} - p_{n+|u'|+f(n)}$, and
- $p_n - p_{|v'|} + 2 \geq p_{n+|u'|+f(n)} - p_{|v'|+|u'|+f(n)}$.

Moreover, from the properties of the sequence p, we get

$$p_{2n+f(n)} \geq 2p_n + 4.$$

Adding the three inequalities above, we obtain

$$p_{|u'|} + p_{|v'|} \leq p_{|v'|+|u'|+f(n)},$$

and as a consequence

$$\#_1 w \leq p_{|v'|+|u'|+f(n)} = p_{|w|}.$$

This means that $u0^{f(n)}v$ does not contain any forbidden pattern, and as a consequence of Lemma 5.15, this pattern is globally admissible for X_p. We proved that the subshift X_p is f-block gluing, which means (Proposition 5.4) that X_p is f-irreducible. $\square$

5.3.2.6 Sketch of the proof of Theorem 5.12

The idea of the proof for Theorem 5.12 is, for any Π_1-computable number α, to give an algorithm which computes a non-increasing sequence $(\beta_n)_n$ such that the conditions of Lemma 5.23 are verified (ensuring the f-irreducibility) and such that the corresponding bounded density subshift has entropy α. The decidability is ensured by the computability of β. The summability condition on the function f in the statement of the theorem is used to ensure that the sequence of entropies of the subshifts corresponding to the finite sequences $(\beta_n)_{n \geq N}$ for $N \geq 1$ converges towards α. Since this limit is also the entropy of the obtained subshift, the entropy of this subshift is equal to α.

5.3.2.7 Computability of the entropy for bounded density subshift associated to a finite sequence

In the construction, at each step n, the following Lemma 5.24 will be used to compute an approximation of the entropy of the obtained subshift after defining only the n first terms of the sequence, and decide the choice of the next term.

Lemma 5.24. *Let f be a computable integer function, and $K > 0$ some parameter. There exists an algorithm that, given a non-increasing finite sequence $(\beta_k)_{k\leq n}$ of positive rational numbers such that for all k, $\beta_k \leq K$, as input, computes the entropy of $X_{f,\beta}$.*

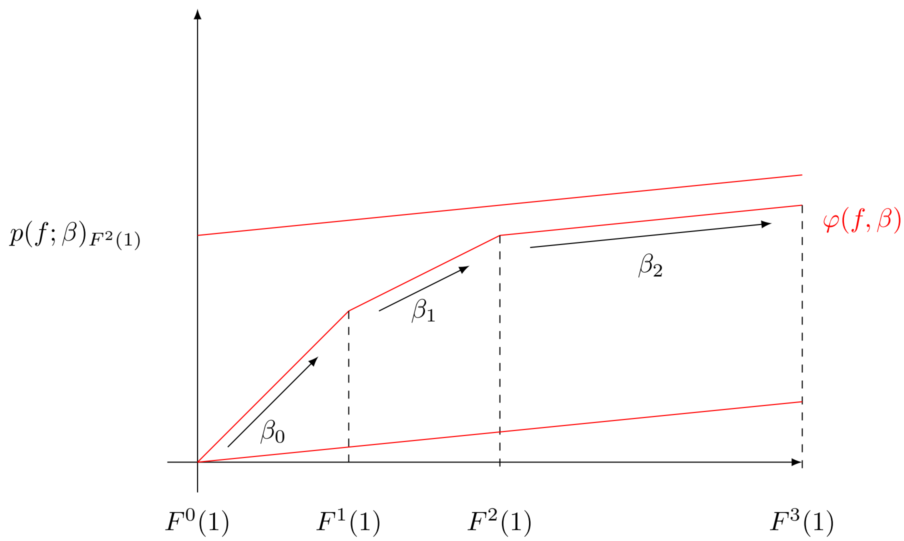


Figure 5.2: Illustration of the proof of Lemma 5.24.

Proof. Let β be a non-increasing sequence of non-negative rational numbers bounded by 1. Since it is non-increasing, the function $\varphi(f,\beta)$ is concave. As a consequence, $p(f,\beta)$ is concave. Since β is non-increasing, for all $k \geq 1$,

$$p(f,\beta)_{F^n(1)} + \left\lceil \beta_n.k \right\rceil \geq p(f,\beta)_k \geq \left\lceil \beta_n.k \right\rceil,$$

as illustrated schematically on Figure 5.2.

As a consequence, for all k,

$$\begin{array}{rcl} p(f,\beta)_{2k+c_n} & \geq & \left\lceil \beta_n.(2k+c_n) \right\rceil \geq 2\beta_n.k + \beta_n.c_n \geq 2(\beta_n.k+1) + 2p(f,\beta)_{F^n(1)} + 4 \\ & \geq & 2p(f,\beta)_{F^n(1)} + \left\lceil \beta_n.k \right\rceil + 4 \geq 2p_k + 4 \end{array},$$

where

$$c_n = \frac{2p(f,\beta)_{F^n(1)} + 6}{\beta_n}.$$

By Lemma 5.23, X_p is c_n-irreducible, and c_n is computable from the knowledge of β and f.

This sequence is also computable, since f and β are computable. It follows, by Lemma 5.16, that $X_{f,\beta}$ is decidable, and one can construct the Turing machine which decides its language from the knowledge of β.

The Lemma follows by application of Theorem 5.9. □

5.3.2.8 Controlling the change of entropy

In this section, we give an upper bound on the entropy change when defining the $(N+1)$ th term of the sequence β when the N first terms are already defined, with a condition on the chosen $(N+1)$th value and the previous ones. In the construction, these conditions will be verified asymptotically.

Lemma 5.25. *Let $\beta = (\beta_n)_{n \leq N+1}$ be a finite non-increasing sequence of positive rational numbers, and f a computable integer function. Assume $\frac{2}{3}\beta_N \leq \beta_{N+1} < \beta_N$ and $\varphi(F^N(1)) \geq 3$ (these conditions are imposed so that the number $\kappa(N)$ under is not equal to zero).*

Then we have the following inequality:

$$\boxed{h(X') \leq h(X) \leq h(X') + \frac{\log_2 \kappa(N)}{\kappa(N)}},$$

where

$$\kappa(N) = \min\left(\left\lfloor \frac{\beta_{N+1}}{2(\beta_N - \beta_{N+1})} \right\rfloor, \left\lfloor \frac{\varphi(F^N(1))}{3} \right\rfloor\right),$$

where we use the following notations:

- *X the subshift associated to f and $(\beta_n)_{n \leq N}$*
- *X' the subshift associated to f and $(\beta_n)_{n \leq N+1}$.*
- *φ the function $\varphi(f, (\beta_n)_{n \leq N})$*

Proof. In this proof, we use the following notations:

- φ' is the function $\varphi(f, (\beta_k)_{k \leq N+1})$.
- p is the sequence $p(f, (\beta_n)_{n \leq N})$.

1. **Left-hand inequality:**

   The left-hand inequality comes from $X' \subset X$, since β is non-increasing.

2. **Right-hand inequality:**

   Let us prove the right-hand inequality. In this order, we define a function δ_N which transforms words of $\mathcal{A}^{\kappa(N)^*}$ in the language of X into words in the language of X'. We evaluates, for each of these image words, the number of its pre-images. This provides an upper bound on the number of words in $\mathcal{L}_m(X)$ which as a function of $|\mathcal{L}_m(X')|$. We deduce then the right-hand inequality.

   (a) **Definition of the function δ_N:**

   This function $\delta_N : \left(\mathcal{A}^{\kappa(N)}\right)^* \to \left(\mathcal{A}^{\kappa(N)}\right)^*$ is defined as follows. For $w \in \mathcal{A}^{m\kappa(N)}$, and j between 1 and k, define $\delta_N(w)$ by $\delta_N(w)_{\ell_j} = 0$ where ℓ_j is the first position in the interval $[\![j\kappa(N), (j+1)\kappa(N) - 1]\!]$ (from left to right) such that the symbol of the word w in this position is 1, and $\delta_N(w)_l = w_l$ for the other positions l. See Figure 5.3 for an illustration.

   (b) **The words in $\mathcal{L}(X)$ are transformed into words in $\mathcal{L}(X')$:**

   Let $w \in \mathcal{L}(X) \bigcap \left(\mathcal{A}^{\kappa(N)}\right)^*$, and denote m such that $w \in \mathcal{L}(\Sigma) \bigcap \left(\mathcal{A}^{\kappa(N)}\right)^m$. We will prove that $\delta_N(w) \in \mathcal{L}(X')$. Using Lemma 5.15, it is sufficient to prove that this word is locally admissible for X'.

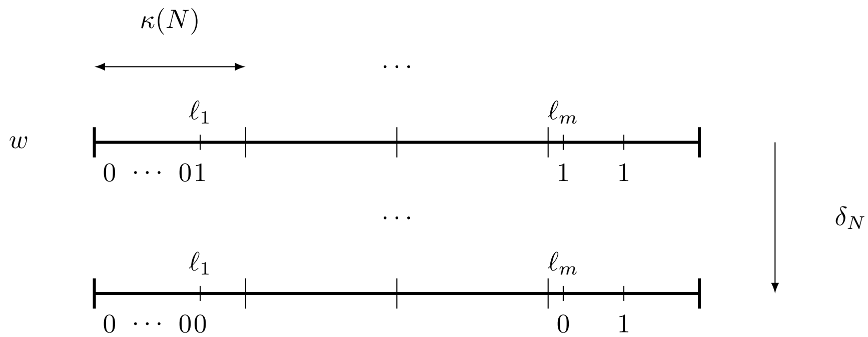


Figure 5.3: Illustration of the definition of the function δ_N.

- **Conditions to verify:**
The conditions left to prove are related to length $F^N(1) + t$ subwords, with $t > 0$. Indeed, the sequences $p(f, (\beta_n)_{n \leq N})$ and $p(f, (\beta_n)_{n \leq N+1})$ agree on their first $F^N(1)$ values and that the number of 1 symbols in any subword is lowered by the application of δ_N.

This is sufficient to prove that for any $t > 0$ and any length $F^N(1) + t$ subword,
  i. either the number of 1 symbols in this subword is already smaller or equal to
$$\left\lceil \varphi(F^N(1) + t) \right\rceil,$$
  ii. or the function δ_N suppresses at least a number
$$\left\lceil (\beta_N - \beta_{N+1}).t \right\rceil$$
  symbols 1 in this subword.

Indeed, in the second case, since w is in the language of X, the number of 1 in its subword is smaller or equal to
$$\left\lceil \varphi'(F^N(1) + t) \right\rceil.$$
After the application of δ_N, the corresponding subword contains at most
$$\left\lceil \varphi(F^N(1) + t) \right\rceil - \left\lceil (\beta_N - \beta_{N+1}).t \right\rceil$$
symbols 1, which is *strictly* smaller than
$$\varphi(F^N(1) + t) + 1 - (\beta_N - \beta_{N+1}).t \leq \varphi'(F^N(1) + t) + 1,$$
thus smaller or equal to
$$\left\lceil \varphi'(F^N(1) + t) \right\rceil,$$
since this number is an integer.
- **Pick a subword of w having enough 1 symbols:**
The word w is written $w = w^1 ... w^m$ where the w^i are length $\kappa(N)$ words.

Let z be a length $F^N(1)+t$ subword of w containing at least

$$\left\lceil \varphi'(F^N(1)+t) \right\rceil$$

symbols 1. We shall prove that the number of 1 symbols suppressed in this word by the function δ_N is at least

$$\left\lceil (\beta_N - \beta_{N+1}).t \right\rceil.$$

This subword is written $uw^k...w^jv$, where $k \leq j$, u is a suffix of w^{k-1} and v is a prefix of w^{j+1}.

- **An inequality about $\kappa(N)$:**
Let us prove that

$$\kappa(N) \leq \frac{\left\lceil \varphi'(F^N(1)+t) \right\rceil - 2\kappa(N)}{\left\lceil (\beta_N - \beta_{N+1}).t \right\rceil}.$$

We have

$$\kappa(N) = \min\left(\left\lfloor \frac{\beta_{N+1}}{2(\beta_N - \beta_{N+1})} \right\rfloor, \left\lfloor \frac{\varphi(F^N(1))}{3} \right\rfloor \right),$$

and as a consequence

$$\kappa(N) \leq \frac{\varphi(F^N(1)) + \beta_{N+1}.t - 2\kappa(N)}{\left\lceil (\beta_N - \beta_{N+1}).t \right\rceil},$$

Indeed,

i. when $t \leq \frac{1}{\beta_N - \beta_{N+1}}$, we have $\left\lceil (\beta_N - \beta_{N+1}).t \right\rceil = 1$, and since $\kappa(N) \leq \frac{\varphi(F^N(1))}{3}$, $\kappa(N) \leq p_{F^N(1)} - 2\kappa(N)$, and then

$$\kappa(N) \leq \frac{\varphi(F^N(1)) - 2\kappa(N)}{\left\lceil (\beta_N - \beta_{N+1}).t \right\rceil} \leq \frac{p_{F^N(1)} - 2\kappa(N) + \beta_{N+1}.t}{\left\lceil (\beta_N - \beta_{N+1}).t \right\rceil}.$$

ii. when $t > \frac{1}{\beta_N - \beta_{N+1}}$, we have

$$2(\beta_N - \beta_{N+1}).t \geq (\beta_N - \beta_{N+1}).t + 1 \geq \left\lceil (\beta_N - \beta_{N+1}).t \right\rceil.$$

Since

$$\kappa(N) \leq \frac{\beta_{N+1}}{2(\beta_N - \beta_{N+1})} = \frac{\beta_{N+1}.t}{2(\beta_N - \beta_{N+1}).t},$$

then

$$\kappa(N) \leq \frac{\varphi(F^N(1)) - 2\kappa(N) + \beta_{N+1}.t}{\left\lceil (\beta_N - \beta_{N+1}).t \right\rceil}.$$

Thus

$$\kappa(N) \leq \frac{\left\lceil \varphi'(F^N(1)+t) \right\rceil - 2\kappa(N)}{\left\lceil (\beta_N - \beta_{N+1}).t \right\rceil},$$

- **Lower bound for the number of 1 symbols suppressed:**
  The number of 1 symbols suppressed by δ_N in the subword is equal to the number of words amongst the words w^i, u and v containing at least one symbol 1.
  The word z contains at least
  $$\left\lceil \varphi'(F^N(1)+t) \right\rceil$$
  symbols 1. Hence the number of words amongst the words w^i, u and v containing at least one symbol 1 is greater than
  $$\frac{\left\lceil \varphi'(F^N(1)+t) \right\rceil - 2\kappa(N)}{\kappa(N)}.$$
  Indeed, if this was not true the maximal possible number of 1 in z would be smaller than
  $$\left\lceil \varphi'(F^N(1)+t) \right\rceil - 2\kappa(N) + 2\kappa(N) = \left\lceil \varphi'(F^N(1)+t) \right\rceil,$$
  taking into account that any of the w^i contains at most $\kappa(N)$ symbols 1, as well as u and v.
  As a consequence of this and the inequality on $\kappa(N)$ proved in the last point, the number of 1 suppressed in z is at least
  $$\frac{\left\lceil \varphi'(F^N(1)+t) \right\rceil - 2\kappa(N)}{\kappa(N)} \geq \left\lceil (\beta_N - \beta_{N+1}).t \right\rceil$$

(c) **Entropy inequality:**
From the definition of δ_N, for all m, a word in $\mathcal{L}(X') \bigcap \mathcal{A}^{m\kappa(N)}$ has at most $\kappa(N)^m$ pre-images. This means that
$$\kappa(N)^m. \left| \mathcal{L}(X') \bigcap \mathcal{A}^{m\kappa(N)} \right| \geq \left| \mathcal{L}(X) \bigcap \mathcal{A}^{m\kappa(N)} \right|.$$
As a consequence,
$$\frac{m.\log_2(\kappa(N))}{m\kappa(N)} + \frac{\log_2 |\mathcal{L}(X') \bigcap \mathcal{A}^{m\kappa(N)}|}{m\kappa(N)} \geq \frac{\log_2 |\mathcal{L}(X) \bigcap \mathcal{A}^{m\kappa(N)}|}{m\kappa(N)},$$
and taking $m \to +\infty$,
$$h(X) \leq h(X') + \frac{\log_2(\kappa(N))}{\kappa(N)}.$$

□

5.3.2.9 Description of the algorithm

Let α be a Π_1-computable non-negative real number and $(\alpha_n)_n$ a non-increasing computable sequence of rational numbers such that $\alpha_n \searrow \alpha$. We define an algorithm that generates a sequence $(\beta_n^\alpha)_{n\in\mathbb{N}}$ such that the associated bounded density subshift X_{f,β^α}, denoted X_α, satisfies $h(X_\alpha) = \alpha$.

It works as follows:

- $\beta_0^\alpha = 4$ (the minimal value ensuring that the irreducibility condition is valid at least for one value in the second point at the first step).

- Let n be some positive integer and assume $\beta_0^\alpha, \dots, \beta_n^\alpha$ are defined. Compute some values $q_n(k) \in \mathbb{Q}$ for all $k \in [\![1, 2^n]\!]$, such that
$$|h(\Sigma_k^{(n)}) - q_n(k)| \leq 2^{-n},$$
where $\Sigma_k^{(n)}$ is the bounded density subshift associated to the function f and the finite sequence $(\eta_k)_{k \leq n+1}$ obtained by putting $\eta_{n+1} = \frac{k}{2^n}\beta_n$ and $\eta_k = \beta_k^\alpha$ for all $k \leq n$; this is possible by using Lemma 5.24. Then put $\beta_{n+1}^\alpha = \frac{k_n}{2^n}\beta_n^\alpha$, with k_n the smallest k such that:

  Entropy condition $q_n(r) \geq \alpha_n + 2^{-n}$

  Irreducibilit condition $p(f, (\eta_k)_{k \leq n+1})_{F^{n+1}(1)} \geq 2p(f, (\eta_k)_{k \leq n+1})_{F^n(1)} + 4.$

See an illustration on Figure 5.4.

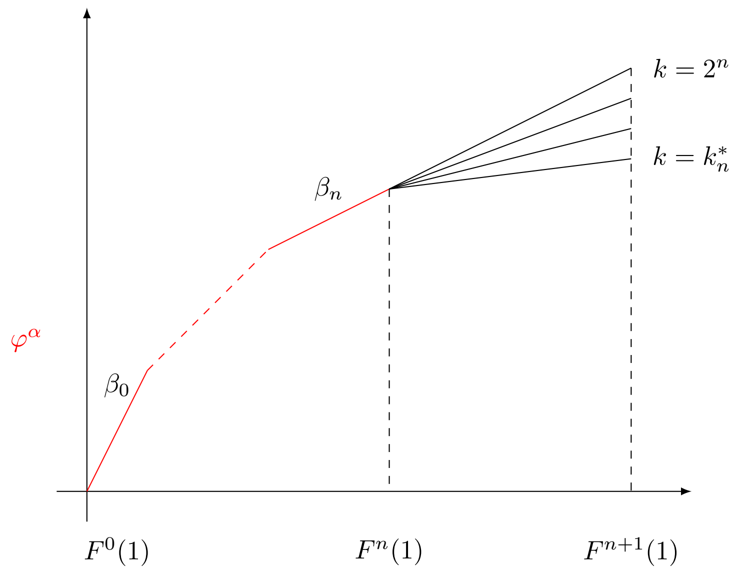


Figure 5.4: Illustration of the definition of the algorithm. The number k_m^* is the smallest one such that the mixing condition is verified.

In the following, we denote $X_{m,\alpha}$ the subshift $X_{f,(\beta_k^\alpha)_{k \leq m}}$, and $p^\alpha = p(f, \beta^\alpha)$.

Remark 38. *A consequence of the entropy condition is that for all m, the entropy of the subshift $X_{m,\alpha}$ is greater or equal to $2^{-m} + \alpha_m$.*

5.3.2.10 Proof of Theorem 5.12 for $d = 1$

In this section, we prove the theorem for $d = 1$. What is left to prove is that the entropy of the subshift X_α defined in Section 5.3.2.9 is α: this is stated as Lemma 5.26.

Lemma 5.26. *For any positive real number α which is Π_1-computable, $h(\Sigma_\alpha) = \alpha$.*

Proof of Lemma 5.26. 1. **Lower bound:**

Since $\mathcal{L}_n(X_\alpha) = \bigcap_m \mathcal{L}_n(X_{m,\alpha})$, we have that
$$h(X_\alpha) = \inf_n \frac{\log_2 |\mathcal{L}_n(X_{f,\beta})|}{n} = \inf_n \inf_m \frac{\log_2 |\mathcal{L}_n(X_{m,\alpha})|}{n} = \inf_m h(X_{m,\alpha}).$$

Since for all m, $h(X_{m,\alpha}) \geq \alpha_m + 2^{-m} \geq \alpha$, by the entropy condition of the algorithm,

$$h(X_\alpha) \geq \alpha.$$

2. **Upper bound:**

   For the sake of contradiction, assume that

   $$h(X_\alpha) > \alpha.$$

   (a) **Convergence of the sequences p^α and β^α.**

   Since $h(X_\alpha) > \alpha$, then $h(X_\alpha) > 0$. From Lemma 5.17, and the fact that p^α is non-decreasing, we get that $p^\alpha_n \to +\infty$. Moreover, from Lemma 5.19, we get that β^α converges towards some positive number.

   (b) **There exists some m^* such that for all $m \geq m^*$, we have the following inequality:**

   $$\frac{\beta^\alpha_m}{2^m} + \frac{p_{F^m(1)} + 4}{F^m(1) + f(F^m(1))} > \beta^\alpha_{m+1} \geq \frac{p_{F^m(1)} + 4}{F^m(1) + f(F^m(1))}.$$

   In order to prove this, for all m there is a unique integer k, denoted k^*_m such that $\frac{k}{2^m}\beta^\alpha_m$ is in the interval

   $$\left[\!\left[\frac{\beta^\alpha_m}{2^m} + \frac{p_{F^m(1)} + 4}{F^m(1) + f(F^m(1))}, \frac{p_{F^m(1)} + 4}{F^m(1) + f(F^m(1))}\right]\!\right].$$

   As a consequence, it is sufficient to prove that there exists m^* such that for all $m \geq m^*$, β_{m+1} is defined to be $\frac{k}{2^m}\beta_m$, for k minimal such that the irreducibility condition is verified at step m. This means that both the entropy and irreducibility conditions are valid for this value of k. We denote for all m,

   $$\beta^*_{m+1} = \frac{k^*_m}{2^m}\beta^\alpha_m.$$

   The steps for this proof are the following:

   i. There exists some m_0 such that for all $m \geq m_0$,

   $$|h(X_{m,\alpha}) - h(X_\alpha)| < \frac{h(X_\alpha) - \alpha}{8},$$

   $$|2^{-m} + \alpha_m - \alpha| < \frac{h(X_\alpha) - \alpha}{4}$$

   ii. There exists some $n_0 > 2$ such that for all $n \geq n_0$,

   $$\frac{\log_2(n)}{n} < \frac{h(X_\alpha) - \alpha}{8}.$$

   iii. Since p^α converges towards $+\infty$,

   $$\left\lfloor \frac{\varphi(f, \beta^\alpha)(F^m(1))}{3} \right\rfloor \to +\infty.$$

   As a consequence, there exists some m_1 such that for all $m \geq m_1$,

   $$\left\lfloor \frac{\varphi(f, \beta^\alpha)(F^m(1))}{3} \right\rfloor \geq n_0.$$

iv. There exists some m_2 such that for all $m \geq m_2$,

$$\beta^*_{m+1} \geq \frac{2}{3}\beta_m.$$

Indeed, since $f(n) = o(n)$, there exists m_2 such that for all $m \geq m_2$,

$$\frac{f(F^m(1))}{F^m(1)} \leq \frac{1}{2}.$$

Then we have

$$\frac{1}{1 + \frac{f(F^m(1))}{F^m(1)}} \geq \frac{2}{3}.$$

Since for all m,

$$\beta^*_{m+1} \geq \frac{p^\alpha_{F^m(1)} + 4}{F^m(1) + f(F^m(1))} \geq \frac{p^\alpha_{F^m(1)}}{F^m(1)} \frac{1}{1 + \frac{f(F^m(1))}{F^m(1)}},$$

and

$$\frac{p^\alpha_{F^m(1)}}{F^m(1)} \geq \beta^\alpha_m,$$

then for all $m \geq m_2$,

$$\beta^*_{m+1} \geq \frac{2}{3}\beta^\alpha_m.$$

v. As a consequence, for all $m \geq \max(m_1, m_2)$, either $\beta^\alpha_{m+1} = \beta^\alpha_m$, else all the conditions of Lemma 5.25 are verified. In this last case, we have

$$|h(X_{m+1,\alpha}) - h(X_{m,\alpha})| \leq \frac{\log_2(n)}{n}$$

for some $n \geq n_0$ (which depends on m), and as a consequence,

$$|h(\Sigma_{m+1,\alpha}) - h(X_{m,\alpha})| \leq \frac{h(X_\alpha) - \alpha}{8},$$

and thus for all $m \geq m^* = \max(m_0, m_1, m_2)$,

$$|h(\Sigma_{m+1,\alpha}) - h(X_\alpha)| \leq \frac{h(X_\alpha) - \alpha}{4},$$

and

$$|2^{-m} + \alpha_m - \alpha| < \frac{h(X_\alpha) - \alpha}{4}.$$

This implies that

$$h(X_{m+1,\alpha}) > \alpha + 2^{-m},$$

and that the entropy condition is verified for the choice $k = k^*_m$.

This implies that for all $m \geq m_*$, $\beta^\alpha_{m+1} = \beta^*_{m+1}$, and then the aimed inequality.

(c) **Contradiction:**

It follows from last point that for $m \geq m^*$, since

$$p^\alpha_{F^{m+1}(1)} \leq p^\alpha_{F^m(1)} + \beta^\alpha_{m+1}.(F^m(1) + f(F^m(1))) + 1,$$

that

$$
\begin{aligned}
p^{\alpha}_{F^{m+1}(1)} &\leq 2p^{\alpha}_{F^m(1)} + 5 + \frac{\beta^{\alpha}_m}{2^m}(F^m(1) + f(F^m(1))) \\
&\leq 2p^{\alpha}_{F^m(1)} + 5 + \frac{1}{2^m} p^{\alpha}_{F^m(1)} \left(1 + \frac{f(F^m(1))}{F^m(1)}\right) \\
&\leq 2\left(1 + \frac{1}{2^m}\right) p^{\alpha}_{F^m(1)} + 5 + \frac{1}{2^m}
\end{aligned}
$$

since $f(F^m(1)) < F^m(1)/2$ for all $m \geq m^*$. This implies that

$$
\begin{aligned}
\frac{p^{\alpha}_{F^{m+1}(1)}}{F^{m+1}(1)} &\leq \left(1 + \frac{1}{2^m}\right) \frac{p^{\alpha}_{F^m(1)}}{F^m(1)} \frac{1}{1 + \frac{f(F^m(1))}{2F^m(1)}} + \frac{5}{F^{m+1}(1)} + \frac{1}{2^m} \\
&\leq \left(1 + \frac{1}{2^m}\right) \frac{p^{\alpha}_{F^m(1)}}{F^m(1)} \left(1 - \frac{f(F^m(1))}{2F^m(1)}\right) + \frac{5}{F^m(1)} + \frac{1}{2^m}
\end{aligned}
$$

Thus

$$
\frac{p^{\alpha}_{F^m(1)}}{F^m(1)} - \frac{p^{\alpha}_{F^{m+1}(1)}}{F^{m+1}(1)} \geq \frac{p^{\alpha}_{F^m(1)}}{F^m(1)} \left(1 + \frac{1}{2^m}\right) \frac{f(F^m(1))}{F^m(1)} - \frac{5}{F^m(1)} - \frac{1}{2^m} \frac{p^{\alpha}_{F^m(1)}}{F^m(1)} - \frac{1}{2^m}.
$$

Applying this result inductively, we get for every $k \geq 1$:

$$
\begin{aligned}
\frac{p^{\alpha}_{F^m(1)}}{F^m(1)} - \frac{p^{\alpha}_{F^{m+k}(1)}}{F^{m+k}(1)} &\geq \left(\inf_n \frac{p^{\alpha}_n}{n}\right) \sum_{i=m}^{m+k-1} \left(1 + \frac{1}{2^i}\right) \frac{f(F^i(n))}{F^{i+1}(n)} - \sum_{i=m}^{m+k-1} \left(\frac{5}{F^i(n)} + \frac{1}{2^i} \frac{p^{\alpha}_{F^i(n)}}{F^i(n)} + \frac{1}{2^i}\right) \\
&\geq 2\left(\inf_n \frac{p^{\alpha}_n}{n}\right) \sum_{i=m}^{m+k-1} \frac{f(F^i(n))}{F^{i+1}(n)} - \sum_{i=m}^{m+k-1} \left(\frac{5}{F^i(n)} + \frac{1}{2^i} \frac{p^{\alpha}_{F^i(n)}}{F^i(n)} + \frac{1}{2^i}\right).
\end{aligned}
$$

Since the sequence $(p^{\alpha}_n/n)_n$ is bounded, the left-hand side is bounded. By Lemma 5.17, $\inf_n \frac{p^{\alpha}_n}{n} > 0$. It follows that the series $\sum_{i\geq 1} \frac{f(F^i(n))}{F^{i+1}(n)}$ converges.
For all n, $f(n) \leq (\beta^{\alpha}_0 + 1)n$. Since f is also nondecreasing, for all i:

$$
\begin{aligned}
\sum_{k=F^{i-1}(1)+1}^{F^i(1)} \frac{f(k)}{k^2} &\leq (F^i(1) - F^{i-1}(1)) \cdot \frac{f(F^i(1))}{(F^{i-1}(1))^2} \\
&\leq \frac{(F^i(1) - F^{i-1}(1)) \cdot F^{i+1}(1)}{(F^{i-1}(1))^2} \cdot \frac{f(F^i(1))}{F^{i+1}(1)} \\
&\leq (\beta^{\alpha}_0 + 2)(\beta^{\alpha}_0 + 3)^2 \frac{f(F^i(1))}{F^{i+1}(1)}
\end{aligned}
$$

Since the series $\sum_i \frac{f(F^i(n))}{F^{i+1}(n)}$ converges, it follows that $\sum_k^{\infty} \frac{f(k)}{k^2}$ as well, a contradiction with the initial hypothesis.

As a consequence,

$$h(X_{p^{\alpha}}) \leq \alpha,$$

which ends the proof.

□

The statement of the theorem follows for $d = 1$ from Lemma 5.16 and Lemmas 5.23 and 5.26.

5.3.2.11 Proof for $d \geq 1$

In order to obtain the same result in higher dimension, notice that for any one-dimensional subshift X, the subshift

$$X^d = \{x \in \mathcal{A}^{\mathbb{Z}^d} : \forall \mathbf{j} \in \mathbb{Z}^{d-1},\ (x_{i,\mathbf{j}})_{i\in\mathbb{Z}} \in X\}$$

has the same entropy and decidability properties as X. Moreover, if X is f-irreducible, then Σ^d is also f-irreducible. Indeed, let $\mathbb{U}, \mathbb{V}$ be two finite subsets of $\mathbb{Z}^d$, u, v two patterns on $\mathbb{U}, \mathbb{V}$ respectively, such that $\delta(\mathbb{U}, \mathbb{V}) \geq f(\max(\delta(\mathbb{U}), \delta(\mathbb{V})))$. For all $\mathbf{k} \in \mathbb{Z}^{d-1}$, consider the sets $\mathbb{U}_{\mathbf{k}}$, $\mathbb{V}_{\mathbf{k}}$ defined to be the intersection of $\mathbb{U}, \mathbb{V}$ with the set of vectors such that the $d-1$ last coordinates are those of $\mathbf{k}$. Denote u_k, v_k the restrictions of the respective patterns u, v on $\mathbb{U}_{\mathbf{k}}$, $\mathbb{V}_{\mathbf{k}}$. Since f is non-increasing, that

$$\delta(\mathbb{U}_{\mathbf{k}}, \mathbb{V}_{\mathbf{k}}) \geq \delta(\mathbb{U}, \mathbb{V})$$

and

$$\max(\delta(\mathbb{U}), \delta(\mathbb{V})) \geq \max(\delta(\mathbb{U}_{\mathbf{k}}), \delta(\mathbb{V}_{\mathbf{k}})),$$

we have

$$\delta(\mathbb{U}_{\mathbf{k}}, \mathbb{V}_{\mathbf{k}}) \geq f(\max(\delta(\mathbb{U}_{\mathbf{k}}), \delta(\mathbb{V}_{\mathbf{k}}))).$$

This implies that there exists some x_k in X such that the restrictions of this configuration on $\mathbb{U}_{\mathbf{k}}$ and $\mathbb{U}_{\mathbf{k}}$ are respectively u_k and v_k.

Let x be the configuration such that the restriction to $\{(i, \mathbf{k}) : i \in \mathbb{Z}\}$ is $x_{\mathbf{k}}$ for all $\mathbf{k} \in \mathbb{Z}^{d-1}$.

The restriction of this configuration to $\mathbb{U}$ and $\mathbb{V}$ are the patterns u, v.

5.4 Comments

1. Let us notice that we show that the *concavity property* is involved in the study of mixing-like properties subshifts.

2. We still don't know if every computable number is the entropy of some f-irreducible subshift with f under the threshold. We conjecture the following.

Conjecture 1. *For any computable non-negative real number, there exists some f-irreducible subshift whose entropy is this number, when f is an integer function such that $\sum_{n=1}^{+\infty} \frac{f(n)}{n^2} < +\infty$.*

In this conjecture, we do not impose that the series converges at computable rate: the intuition tells that this condition is artificial and could be suppressed from the theorems.

Chapter 6

Entropy dimensions of minimal multidimensional SFT

Sommaire

This chapter is a partial reproduction of the article *A characterization of the entropy dimensions of minimal $\mathbb{Z}^3$-SFTs* [GS17b].

Abstract

We prove in this chapter that the possible entropy dimensions of minimal $\mathbb{Z}^3$-SFT are the Δ_2-computable numbers in $[0, 2]$, using Goldbach's theorem on Fermat numbers.

6.1 Introduction

In Chapter 4 and Chapter 5, we investigated the influence of the block gluing properties on computational properties of SFT.

Although the possible entropies of a $\mathbb{Z}^d$-subshift for all $d \geq 1$ are all the Π_1-computable numbers, the entropy of a minimal SFT is always zero. One could think that this prevents the embedding of universal computation in these subshifts. However, the study of other invariants of these subshifts reveals that this class of subshifts is also rich.

For instance, in [HV17], the authors proved that minimal subshifts can exhibit complex computational behaviors, by exhibiting minimal $\mathbb{Z}^d$-subshifts that have complex Turing spectrum. In [JLKP17], the authors provided a construction that allows the realization of all the non-negative real numbers smaller than 2 as entropy dimension of minimal $\mathbb{Z}^2$-subshifts (which are not though subshifts of finite type).

To our knowledge, the only other constructions of minimal SFT are due to B. Durand and A. Romashchenko [DR17b]. They use a fixed-point construction in which are implemented computing machines that check if no forbidden pattern in a recursively enumerable set appears in bi-infinite words on the alphabet $\{0,1\}$. Since in these constructions the computation areas of the machines are sparse, the degenerated behaviors are simple. Controlling the growth of the computing units, they could attribute the function of some particular sub-units in order to simulate these behaviors in any computing unit. The minimality of the architecture used for the control on the apparition of the forbidden words is ensured by this simulation.

The idea of simulation is present in both constructions. However, the construction of [DR17b] relies on a very rigid architecture. We propose here a more flexible way to ensure the minimality, although complex to formulate. This allows the simulation of a consequent set of patterns.

In this text, we prove that the numbers that are the entropy dimension of a $\mathbb{Z}^3$-SFT are the Δ_2-computable numbers in $[0,2]$. Thus, there is a difference with the set of numbers that are the entropy of $\mathbb{Z}^3$-SFT, the Δ_2-computable numbers in $[0,3]$.

The construction used to prove the realization part of this characterization is an adaptation of the construction presented in [Mey11]. This construction is not minimal for the reason that pathological behaviors of the Turing machines can appear, and that it uses a display of random bits that breaks the minimality property.

As in Chapter 4, there is an attractive analogy between counters used in the construction and DNA. This comes from the separation into coding part and non-coding one. This analogy suggests that the non-coding part is implied in global properties of a living system.

The structure layer used in this construction is a $3d$ version of the Robinson subshift constructed with three copies of the rigid version of this subshift presented in Chapter 3. It exhibits similar hierarchical structures as the two-dimensional version. Although we don't prove how here, this structure allows, by adding colors, some structures used in [Hoc09] and [PS15] in order to construct $\mathbb{Z}^3$-SFTs to be recovered.

Let us also remark that some of the mechanisms described in the construction presented in this text can be reformulated using substitutions and S-adic systems. However, we don't use Mozes theorem (proved in [Moz89]) or the result proved in [AS14]. (this theorem states that multidimensional S-adic systems are sofic) for two reasons. The first one is that some of the substitution mechanisms are localized in restricted parts in each configuration. When the mechanism is global, there is still an obstacle for the use of these theorems: the need of more precise properties on the SFT than stated in these theorems. One of these properties is the repetition, in each configuration of the SFT, of patterns whose size some power of two, with period equal as well to some power of two. We instead use directly similar techniques as [Moz89] in the construction.

This Chapter is organized as follows: In section 6.2, we prove that the entropy dimensions of a minimal SFT are smaller than $d-1$. Then in section 6.3, we prove the realization part of the characterization.

6.2 Obstruction

In this section we prove that the entropy dimensions of a minimal SFT are constrained as follows:

Proposition 6.1. *Let d be some positive integer. Let X be a minimal $\mathbb{Z}^d$-SFT. Then $\overline{D_h}(X) \leq d-1$.*

Remark 39. *The proof of this proposition was communicated to us by P. Guillon.*

Proof. **Idea:** *the idea of the proof is to construct an element of the subshift having low complexity. This construction using only the fact that the subshift is of finite type. Since the subshift is minimal, the complexity of the subshift is equal to the complexity of this element. We deduce the upper bound on the entropy dimension.*

Let X be some SFT on alphabet $\mathcal{A}$. Let $r > 0$ be the rank of the SFT.

1. **Definition of the annulus and cross supports:**

   Let us denote, for all $n \geq 0$, $r_n = 2^{n+2}r$, and define

$$\mathbb{O}_n = [\![1, r_n]\!]^d \backslash [\![1+r, r_n - r + 1]\!]^d,$$

$$\mathbb{C}_n = \bigcup_{1 \leq k \leq d} [\![1, r_n]\!]^{k-1} \times [\![r_{n-1} - r + 1, r_{n-1} + r[\![\times [\![1, r_n]\!]^{d-k},$$

   when $n \geq 1$, and $\mathbb{C}_0$ is the set $[\![r+1, 2r]\!]^d$. Informally, we call $\mathbb{O}_n$ the outside of set $[\![1, r_n]\!]^d$, and the inside is the complementary of $\mathbb{O}_n$ in this set.

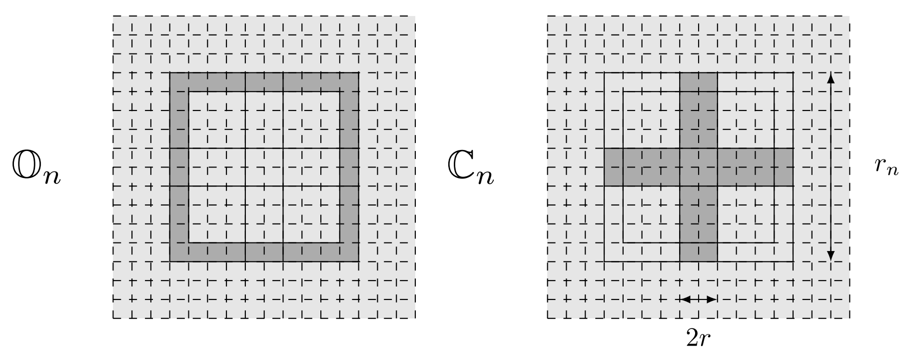


Figure 6.1: Illustration of the definition of the sets $\mathbb{O}_n$ and $\mathbb{C}_n$ (respectively dark gray set on the left and on the right).

2. **An association between patterns on the annuli and patterns on the crosses:**

   Let $\psi_n : \mathcal{L}_{\mathbb{O}_n}(X) \to \mathcal{L}_{\mathbb{C}_n}(X)$ be a function which to some pattern p associates a possible completion of p on $\mathbb{O}_n \bigcup \mathbb{C}_n$.

   Let us construct recursively a function $\varphi_n : \mathcal{L}_{\mathbb{O}_n}(X) \to \mathcal{L}_{[\![1,r_n]\!]^d}(X)$ as follows (See an illustration on Figure 6.2). Consider p a pattern in $\mathcal{L}_{\mathbb{O}_n}(X)$.

(a) Extend it with the pattern $\psi_n(p)$.

(b)
- if $n \geq 1$, consider the restrictions of the obtained pattern on the sets consisting in a product of d elements of $\{[\![1, r_{n-1}]\!], [\![r_{n-1}+1, r_n]\!]\}$. These sets are translate of $\mathbb{O}_{n-1}$. Apply the function φ_{n-1} then the completion on these copies of $\mathbb{O}_{n-1}$.
- if $n = 0$, then the construction is done.

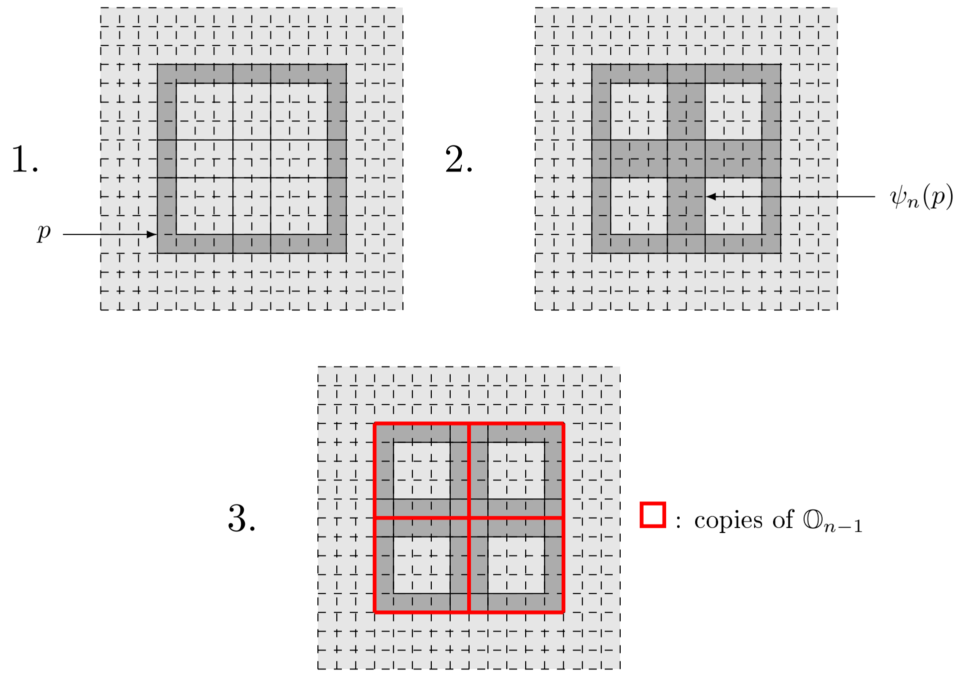


Figure 6.2: Illustration of the construction of the functions φ_n.

3. **Construction of the configuration:**

   Consider the a sequence of patterns $(p_n)_n$ such that for all n, p_n has support $\mathbb{O}_n$, and $(x_n)_n$ a sequence of configurations whose restriction on

$$\left[\!\left[1 - \frac{r_n}{2}, \frac{r_n}{2}\right]\!\right]^d$$

   is equal to $\varphi_n(p_n)$. There exists such a sequence, since the restriction of this pattern on $\mathbb{O}_n$ is globally admissible.

   By compactness, we can extract some infinite sub-sequence of $(x_n)_n$ that converges towards some configuration x of X.

4. **Complexity of the configuration x:**

   Let us fix some $k \geq 1$ and let us give an upper bound of the number of r_k-block that appear in x. Consider p some r_k-block that appear in x, in position $\mathbf{u}$. There exists some $n \geq k$ such that $[\![1 - \frac{r_n}{2}, \frac{r_n}{2}]\!]^d$ contains $\mathbf{u} + \mathbb{C}_k$, and the configuration x_n coincides with x on $\mathbf{u} + \mathbb{C}_k$. As we can decompose $[\![1 - \frac{r_n}{2}, \frac{r_n}{2}]\!]^d$ into copies of $[\![1 - \frac{r_k}{2}, \frac{r_k}{2}]\!]^d$ on which the patterns have their inside

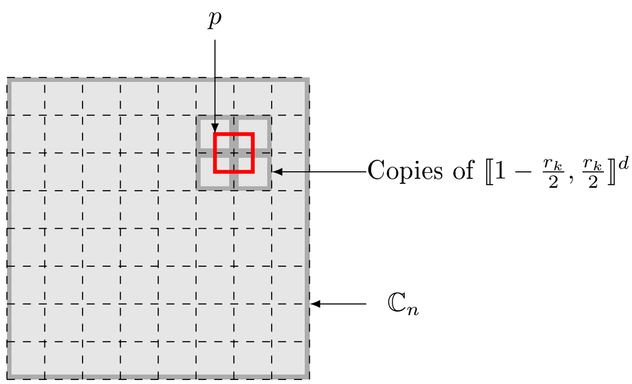


Figure 6.3: Illustration of a typical pattern of the configuration x

determined by their outside, the pattern p is a sub-pattern of a concatenation of 2^d such blocks (See an illustration of Figure 6.3).

Hence the number of r_k blocks in x is smaller than the number of elements of $((\mathcal{A})^{\mathbb{O}_k})^{2^d}$, which is smaller than

$$|\mathcal{A}|^{2^d|\mathbb{O}_k|} \leq |\mathcal{A}|^{2^d*2d*r(r_k)^{d-1}},$$

because $\mathbb{O}_k$ is the union of $2d$ cuboids having dimensions $r, r_k, ..., r_k$ (as illustrated on Figure 6.4).

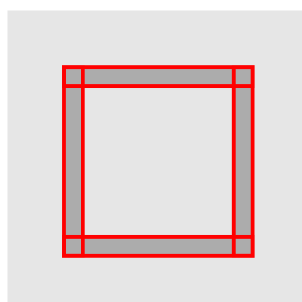


Figure 6.4: The set $\mathbb{O}_k$ can be decomposed into $2d$ cuboids. On this picture, $d = 2$.

Since X is minimal, any globally admissible pattern appears in x. As a consequence,

$$N_{r_k}(X) \leq |\mathcal{A}|^{2^d.2d.r.(r_k)^{d-1}},$$

which means that $\overline{D_h}(X) \leq d-1$.

□

As a consequence of this and of Theorem 3.15, we have the following proposition:

Proposition 6.2. *Let X be some minimal $\mathbb{Z}^3$-SFT admitting an entropy dimension. Then its entropy dimension is a Δ_2-computable number in $[0,2]$.*

6.3 Realization

In this section, we complete the prove following theorem, by proving its realization part:

Theorem 1.4 ([GS17b]). *The possible entropy dimensions of minimal $\mathbb{Z}^3$-SFT are the Δ_2-computable number in $[0, 2]$.*

Before providing an outline of the proof of this theorem, we need some characterization of the Δ_2-computable numbers in terms of Cesaro mean.

6.3.1 Δ_2-computable numbers

Definition 6.3. *A sequence $(a_n)_n \in \{0, 1\}^{\mathbb{N}}$ is said to be Π_1-computable if there exists an algorithm that, taking as input a order pair of integers (n, i), outputs some $\epsilon_{n,i} \in \{0, 1\}$ such that for all n, $a_n = \inf_i \epsilon_{n,i}$.*

The following lemma establishes a link between Δ_2-computable real numbers and Π_1-computable sequences.

Lemma 6.4. *A real number $z \in [0, 2]$ is Δ_2-computable if and only if there exists some Π_1-computable sequence $(a_j)_j$ such that*

$$z = \lim_{n \to \infty} \frac{2}{n} \sum_{j=1}^{n} a_j$$

This lemma is stated as part of Lemma 4.1 in [Mey11].

6.3.2 Abstract of the construction

In this section, we provide an abstract of the proof of Theorem 1.4. We first present the principles of Meyerovitch's construction for the proof of Theorem 3.19, since we use these principles in our construction.

6.3.2.1 Principles of the proof of Theorem 3.19

Let $z \in]0, 2]$ a Δ_2-computable number (0 is easily realized as the entropy dimension of a minimal SFT). Let us describe how to construct a $\mathbb{Z}^2$-SFT whose entropy dimension is z. From Lemma 6.4, there exists some Π_1-computable sequence $(a_n)_n \in \{0, 1\}^{\mathbb{N}}$ such that

$$z = 2 \lim_n \frac{\sum_{k=0}^{n} a_k}{n+1}.$$

This construction relies on the propagation of a signal through the petal hierarchy, which is transformed through the intersections of petals. The signal has two possible colors and . In our construction, we call them hierarchy bits. The transformation rule is abstracted on Figure 6.5. It depends on a bit f_n in $\{0, 1\}$ attached to order n petals. These bits are called frequency bits in our construction.

On the corners of the order 0 petals are superimposed random bits in $\{0, 1\}$. The bits f_n, $n \geq 0$ are imposed to be a_n by a Turing machine using a similar construction as in [HM10]. This architecture is set up so that it does not contribute to the entropy dimension, by using thin computation areas. Thus only the random bits do contribute to the entropy dimension. The number of corners in an order zero petal which is inside an order n cell whose border is colored is equal to

$$4^{\sum_{k=0}^{n} a_k}.$$

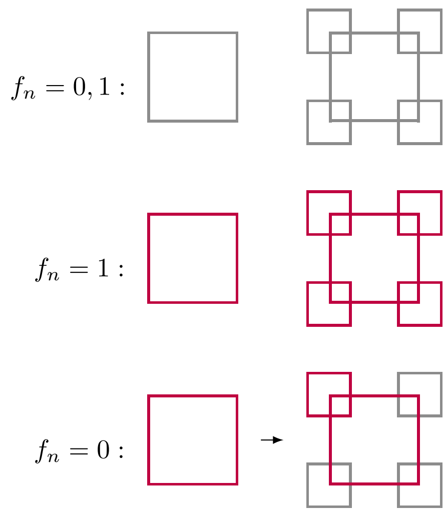


Figure 6.5: Illustration of the signal transformation in Meyerovitch's construction.

This comes from iterating the transformation rule. This number is 0 when the border of the cell is colored .

As a consequence, the number of possible sets of random bits over an order n cell is approximately

$$2^{4^{\sum_{k=0}^{n} a_k}}.$$

Moreover, the number of possibilities for the random bits over a block of the structure having the same size as an order n cell is equal as the number of possibilities for the random bits on such a cell.

Since the entropy dimension is generated by random bits, it follows that the entropy dimension is equal to

$$\lim_n \frac{\log_2(\log_2(2^{4^{\sum_{k=0}^{n} a_k}}))}{\log(2^{n+1})} = 2 \lim_n \frac{\sum_{k=0}^{n} a_k}{n+1} = z,$$

since the size of an order n supertile is approximately 2^{n+1}.

6.3.2.2 Obstacles to the minimality and their solutions

The obstacles to the minimality property in this construction are due to:

1. the definition of the hierarchy bits, since there is some configuration whose hierarchy bits are all (hence the hierarchy bit does not appear);

2. the definition of the random bits, which can be all 0 in a configuration, and all 1 in another one;

3. the machines computations: in the space time diagram of a machine in an infinite computation area, there can appear some parts that never appear in a finite space-time diagram.

In this text we propose a construction, abstracted in the following section, that overcomes as follows these difficulties so that the SFT is minimal:

1. The trick used to overcome the difficulties relative to hierarchy bits is to modify the transformation rules defining these bits. This modification is done so that for a sparse set of levels, no

matter the color of this level, at least one petal below in the hierarchy is colored ■. The set is chosen sparse enough so that this modification does not contribute to the entropy dimension.

2. We use a counter, called hierarchical counter, in order to alternate the possible sets of random bits. For the incrementation, we group the random bits into sets forming independent counters. That is why we need a third dimension in order to realize the numbers in $[0, 2]$, since the display of random bits is bidimensional. The three-dimensional subshifts that we construct use structures that appear in a three-dimensional version of the Robinson subshift.

3. The difficulties coming from the computations are solved by simulating any finite space-time diagram with any initial tape. In order to have the minimality property, we alternate these space-time diagrams in any configuration using counters called linear counters. They code for the initial tape of the machine. Moreover, a machine detecting an error sends an error signal to its initial tape. This error signal is taken into account if and only if the machine was well initialized.

4. The two types of counters have coprime periods for different levels, in order to ensure the minimality. Moreover, they are incremented in orthogonal directions.

On Figure 6.6 is some simplified schema of the construction.

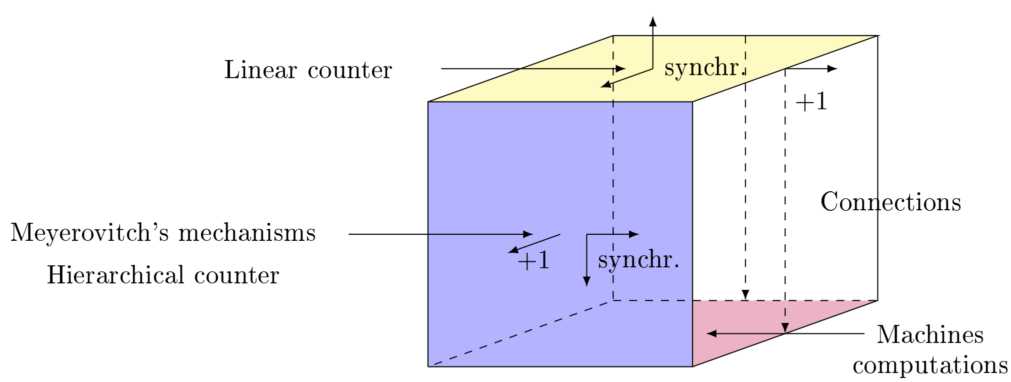


Figure 6.6: Simplified schema of the construction presented in this text. The cube represents a three-dimensional version of the cells observable in the Robinson subshift.

The main arguments in the proof of the minimality property of this subshift are the following ones:

1. Any pattern P can be completed into a pattern P' over a three-dimensional cell with controlled size. Hence it is sufficient to prove that any pattern over a three-dimensional cell appears in any configuration. Such a pattern is characterized by the values of the counters of a sequence of cells included in its support, intersecting all the intermediate levels.

2. One can find back any sequence of values for the counters contained in the three-dimensional cell starting from any cell. This is done in two steps. First by jumping multiple times from a cell to the adjacent one having the same order in the direction of incrementation of the linear counter. Then in the direction of incrementation of the hierarchical counter. This is possible since the periods of the counters are coprime.

This is illustrated on Figure 6.7. On this figure, t (resp. t') is the function which, taking as input the sequence of values of the counters of cells of intermediate level into a cell, outputs the sequence of the adjacent cell in the incrementation direction of the linear (resp. hierarchical) counter.

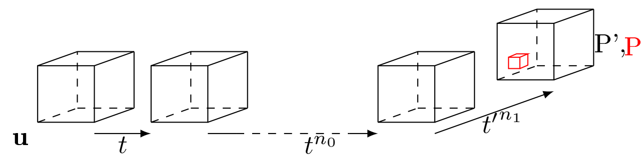


Figure 6.7: Schema of the proof for the minimality property of X_z.

6.3.2.3 Description of the layers

Let z some Δ_2 number in $]0,2]$ and $p = 2^m - 1$ be some *Mersenne* number, for some m such that $m/(2^m - 1) < z/2$. Let us construct some minimal $\mathbb{Z}^3$-SFT X which has entropy dimension z. Using Lemma 6.4, there exists some Π_1-computable sequence $(a_j)_j \in \{0,1\}^{\mathbb{N}}$ such that

$$z = 2 \lim_{n\to\infty} \frac{1}{n+1} \sum_{j=0}^{n} a_j.$$

We assume, without loss of generality, that for all k, $a_{2^k} = 0$. Indeed, this does not change the limit

$$\lim_{n\to\infty} \frac{1}{n+1} \sum_{j=0}^{n} a_j,$$

and the computability properties of the sequence $(a_n)_n$. We also assume that for all $k \in [\![0, p-1]\!]$, there exist infinitely many $i \in \mathbb{N}$ such that $a_{ip+k} = 0$ and infinitely many i such that $a_{ip+k} = 1$.

Let us construct a minimal $\mathbb{Z}^3$-SFT X_z which has entropy dimension $1/p + z(1 - 1/2p)$. This subshift is presented as the superposition of various layers described as follows:

- *Structure layer [Section 6.4.1]:* this layer is a three-dimensional version of the Robinson subshift. We use three superimposed copies of the Robinson subshift, each one of these being constant respectively in the directions $\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3$. We analyze in the corresponding section the properties of this layer. These properties are similar to the ones of the two dimensional version, in terms of supertiles (finite and infinite), repetition of the supertiles, cells, and completion of the patterns into supertiles (ensuring the minimality property). Only the order qp, $q \geq 0$, three-dimensional cells will support functions. The possible functions are the following ones:

  - to execute some Turing computations, in order to control the value of the frequency bits and grouping bits, introduced in the next sections.
  - to increment some counter, whose value consists in a sequence of symbols. There are two type of counters:

    1. *Linear counters*: coding for the initial tape of the Turing machine the direction of propagation of its error signal, the states of the machine heads entering on the sides of the area, and the activity of lines and columns of the area. When the column or line is unactive, this column is not used for computation. This is used so that any part of a space-time diagram can be completed into a space-time diagram on a face of a three-dimensional cell.
    2. *Hierarchical counters*: coding for random bits that are superimposed to the frequency bits, which generate the entropy dimension.

  - to transfer the information contained in the linear counter to the bottom (initial tape) and top line (for the error signal propagation direction) of the machine face, as well as the sides of the face (corresponding the the machine heads entering on the sides of the area).

  Each of these functions is ensured on a specific face of the cell. The separation of the machine function and the counters functions is a necessity for ensuring the minimality property. This appears clearly in the proof of Proposition 6.12, which states that any pattern can be completed into a valid pattern over a (finite) three-dimensional cell.

- *Functional areas layer [Section 6.4.2]:* This layer serves to construct functional areas on the faces of the three-dimensional cells and faces connecting their ridges. This is done in order to localize the possible positions of the counters and machine symbols, and to give a function to these positions (step of computation, vertical or horizontal information transfer), that we call functional positions. This uses a signaling process, similar to a substitution, through hierarchical structures that appear on the faces of the three-dimensional cells and faces connecting these cells.

  See Figure 6.8 for a schema of the functional areas over the faces of a three-dimensional cell.

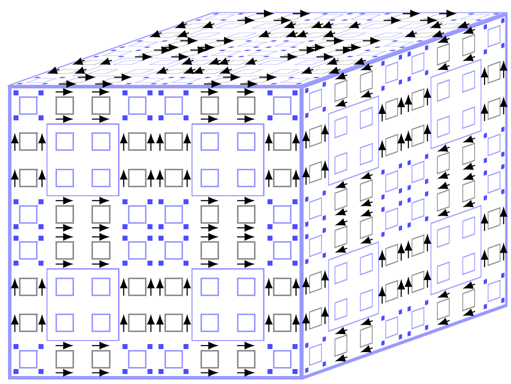


Figure 6.8: Schema of the functional positions on the a face of a three-dimensional cell.

  We then select sub-areas of these functional areas that will support computations of the machines and counters. They are constructed sparse enough so that the values of the linear counter and the computations of the Turing machine do not contribute to the entropy dimension of the subshift (they contribute to the complexity function, but not to the entropy dimension because of Proposition 2.38). Thus the entropy dimension is generated by the values of the hierarchical counter. That is where the number p is used: the selection process relies on the progressive selection of two dimensional cells on the faces according to a horizontal (resp. vertical) address in $\{0,1\}^p$ of these cells relatively to cells higher in the hierarchy. The principle of this address is similar to the coding of a Cantor set: 0 means that the cell is on the left and 1 on the right (resp. on the bottom and on the top for vertical addresses). As a consequence, this process allows the selection of columns (resp. lines) of the functional areas of the faces, through the selection of order 0 cells.

- *Frequency bits layer [Section 6.4.3]:* The frequency bits are bits in $\{0,1\}$ and are attached to each order qp two dimensional cell on their border, in each of the copies of the Robinson subshift. Two cells having the same order have the same bit. The machine can read these bits on its tape (we can make the selection process of functional areas such that we keep this access to the information), and check that the frequency bit of order qp cells is a_q for all $q \geq 0$.

- *Grouping bits layer [Section 6.4.4]:*

  The grouping bits are bits in $\{0,1\}$ that are attached to order qp two-dimensional cells on their border. The difference with frequency bits is their value. When $q = 2^k$ for some k the grouping bit is 1. Else this bit is 0. These bits are imposed by computations of the Turing machines. They are involved in the mechanism that groups the random bits into hierarchical counter values.

- *Linear counter layer [Section 6.4.5]:* This layer supports the value of the linear counter, its incrementation and the transport of the value to the adjacent three dimensional cells having the same order and to the machine face. The linear counter has values only on the three-dimensional cells that have order qp for some $q \geq 0$ (and machines computations are present only on these cells). The incrementation is done on the bottom line of the upper face of the three-dimensional cells

  The value of the counter consists in a word on the alphabet

  $$\mathcal{A} \times \mathcal{Q}^3 \times \{\rightarrow, \leftarrow\} \times \{\mathtt{on}, \mathtt{off}\}^2 \times \mathcal{D},$$

  where $\mathcal{A}$ is the tape alphabet of the machine, $\mathcal{Q}$ is the state set. These two sets are chosen to have cardinality 2^{2^l} for some integer l. This is possible by completing the machine's alphabet and state set by elements that do not interact with the others, in order to not alter the work of the machine on well initialized tape. The set $\mathcal{D}$ a finite set whose cardinality is chosen to be $2^{4.2^l-2}$, so that the counter alphabet has cardinality which is

  $$2^{8.2^l}.$$

  The incrementation mechanism is the one of an adding machine acting on the counter value, except for one step, when the counter value is maximal. When this happen, the action of the adding machine is suspended for one step.

  As a consequence, since the number of columns in the selected sub-areas of the functional areas is a power of 2 and that these numbers are different for two qp order cells, the number of values of linear counters for two different of these levels are numbers $2^{2^{l_1}}$ and $2^{2^{l_2}}$, where $l_1 \neq l_2$. Since the counters are suspended for one step, the periods of the two counters are $2^{2^{l_1}}+1$ and $2^{2^{l_2}}+1$. These are two different, and thus coprime, Fermat numbers.

  The symbols in the set $\{\rightarrow, \leftarrow\}$ codes for the direction of an error signal propagation, The symbols in $\{\mathtt{on}, \mathtt{off}\}$ tell which ones of the lines and columns are used in the area for computations. The error signal is triggered when the machine ends its computation in an error state. This signal is sent to of the ends of the initial tape (according to the propagation direction), in order to verify that the tape was well initialized. We forbid the coexistence of the error signal with a signal that certifies that the tape was well initialized.

- *Machine layer [Section 6.4.6]:*

  This layer supports the computations of the machines. These machines check that the qth frequency bit, shared by qp cells, is equal to a_q. They also check the values of the grouping bits. The initial tape of the machine corresponds to the projection of the linear counter value on $\mathcal{A} \times \mathcal{Q}$, where $\mathcal{A}$ is the tape alphabet and $\mathcal{Q}$ is the state alphabet.

  The machines and the linear counters are implemented in opposite faces of the three-dimensional cells, in order to ensure the separation of the information. This principle allows the minimality property. The information of the linear counter is connected to the machine through signals that propagate in the other faces of the cells.

- *Hierarchical counter layer [Section 6.4.8]:* In this layer some bits in {■,■}, called hierarchy bits, are superimposed to the two-dimensional cells in the copy of the Robinson subshift parallel to the hierarchical counter face of the three-dimensional cells. These bits are determined by a signaling process through the hierarchical structures of any of the copies of the subshifts X_{adR} in the structure layer. This process relies on the frequency bits. On the border of the order $2^k p$ cells, the three hierarchy bits are imposed to be equal.

  On the blue corners having hierarchy bit equal to ■ are superimposed some random bits in $\{0, 1\}$. These bits generate the entropy dimension. As a consequence, the machines have control over the entropy dimension through frequency bits.

  For all k, the kth hierarchical counter value is the set of random bits on positions with blue corners that are in an order $2^k p$ order two-dimensional cell and not in an order $2^j p$ cell, with $j < k$.

  The value of the kth hierarchical counter is incremented on the hierarchical counter face of the three dimensional order $2^k p$ cells, The incrementation mechanism is similar as the one of linear counters, except that it uses discrete curves to represent the value of the counter as a finite sequence.

  This signaling process is done in such a way that the number of hierarchy bits on a face of an order $2^k p$ cell, $k \geq 0$, is strictly growing according to k. Since this number is also a power of 2, the periods of the hierarchical counters of two different levels are different Fermat numbers.

  The direction of incrementation is chosen orthogonal to the incrementation direction of the linear counter. As a consequence, even if a linear counter has the same period as a hierarchical counter, this has no influence on the minimality. On the faces of the other three-dimensional faces, the values of the counters is not changed.

- *Synchronization layer [Section 6.4.9]:* this layer is used to synchronize the hierarchical counters of three-dimensional cells having the same order which are adjacent in the directions that are orthogonal to their incrementation direction. The linear counter is coded to have directly this synchronization.

After that the X_z is constructed for all z, the proof is as follows: take x some Δ_2-computable in $[0, 2]$. If $x = 0$, then any subshift having a unique symbol is minimal and has entropy dimension equal to x. When $x > 0$, for all m such that $1/(2^m - 1) < x$, there exists some z_m such that $1/p + z_m(1 - 1/2p) = x$. We take m such that

$$\frac{x - 1/p}{1 - 1/2p} > \frac{m}{2^m - 1}.$$

For this m, z_m is Δ_2-computable, X_{z_m} is minimal and has entropy dimension equal to x.

Let us make explicit the local rules that induce these global behaviors.

6.4 Details of the construction of the subshifts X_z:

6.4.1 Structure layer

In this section, we describe the construction of the structure layer.

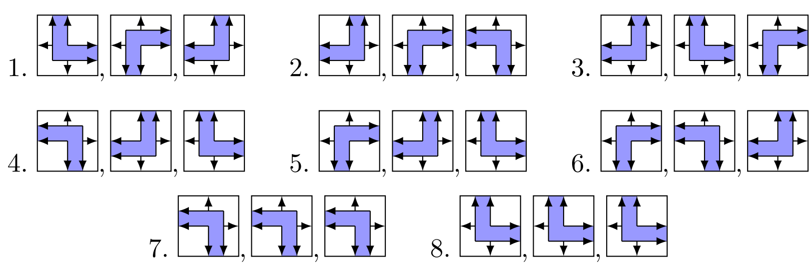
For this purpose, we construct a three dimensional equivalent of the subshift X_{adR}, and analyze this subshift from the point of view of supertile hierarchical structure, repetition of these supertiles, and infinite supertiles.

This subshift consists in the superposition of three copies of X_{adR}, respectively parallel to the vectors ($\mathbf{e}^1$ and $\mathbf{e}^2$), ($\mathbf{e}^2$ and $\mathbf{e}^3$) and ($\mathbf{e}^3$ and $\mathbf{e}^1$).

6.4.1.1 A minimal three-dimensional version of the Robinson subshift

This subshift has *alphabet* $\mathcal{A}^3_{adR}$.

- **Robinson rules in the two dimensional sections of $\mathbb{Z}^3$:** for $\mathbf{i},\mathbf{j} \in \mathbb{Z}^3$ such that $\mathbf{j}-\mathbf{i} = \mathbf{e}^2$, or $\mathbf{e}^3$ (resp. $\mathbf{e}^1$ or $\mathbf{e}^3$, resp. $\mathbf{e}^1$ or $\mathbf{e}^2$), the first (resp. second, resp. third) coordinates of the triples over these positions verify the rules of the subshift X_{adR}. The orientation of the two-dimensional sections of $\mathbb{Z}^3$ where the rules of the subshift X_{adR} are verified is given as follows (the orientation of the Robinson symbols depend on this orientation):
  - for the first coordinate, the horizontal direction is $\mathbf{e}^2$ and the vertical one $\mathbf{e}^3$. This means that when looking in the direction opposite to the vector $\mathbf{e}^3$, and orienting $\mathbf{e}^1$ to the right and $\mathbf{e}^2$ upwards, we see the usual picture of a configuration of X_{adR}.
  - for the second one, the horizontal direction is $\mathbf{e}^3$ and the vertical one $\mathbf{e}^1$.
  - for the last one, the horizontal direction is $\mathbf{e}^1$ and the vertical one $\mathbf{e}^2$.
- **Invariance in the orthogonal direction:** For $\mathbf{i},\mathbf{j} \in \mathbb{Z}^3$ such that $\mathbf{j}-\mathbf{i} = \mathbf{e}^1$ (resp. $\mathbf{e}^2$, resp. $\mathbf{e}^3$) second (resp. first) coordinates of the ordered pairs over these positions are equal. See Figure 6.9 for an illustration.
- **Coincidence rules:** When on some position there are at least two corners symbols of the Robinson subshift, then the three symbols are corners having the same color (blue or red).
- Moreover, the possible triples of blue corners are the following ones:



These triples correspond to the corners of a cube as on Figure 6.10. This cube corresponds to the support of apparition of a pattern whose restriction on each of the coordinates on the corresponding face is a two dimensional order n cell. We call these cubes order n **three dimensional cells**. Notice that the restriction imposed by these rules is not trivial, since the number of allowed triples of blue corners is 8 which is less than the total number of possibilities, which is 4^3. There is an equivalent restriction on triples of red corners.

- When on a position there is only one corner, then the ordered pair of other symbols is amongst the following types:
  - (1) Two six arrows symbols or two five arrows symbols, pointing in the same direction, and orthogonal (in $\mathbb{Z}^3$) to the corner: this type of triples corresponds to the center of the faces of the cubes.
  - (2) Two four arrows symbols or two tree arrows symbols, pointing in the same direction and orthogonal (in $\mathbb{Z}^3$) to the corner. This type corresponds to the edges of the cubes and to the edges of connecting their corners.
  - (3) Two six arrows symbols or two five arrows symbols, pointing in the opposite directions of the arms of the corner. This type corresponds to the centers of the ridges.

– (4) Any ordered pair of four or three arrows symbols that are orthogonal (in $\mathbb{Z}^3$), and parallel to the corner. This type corresponds to the **internal faces** of the cubes.

The intersection of the cell with the $\mathbb{Z}^2$-section of $\mathbb{Z}^3$ that cut a three dimensional cell in two equal parts is called an **internal face** of the cell.

- The triples with only arrows symbols which are two by two orthogonal (in $\mathbb{Z}^3$) are forbidden (the other triples of arrows symbols correspond to the ridges of the internal faces, or the inside of these faces).

- On any translate of the subset $\mathbb{U}_2^3$ of $\mathbb{Z}^3$, there is an admissible blue triple.

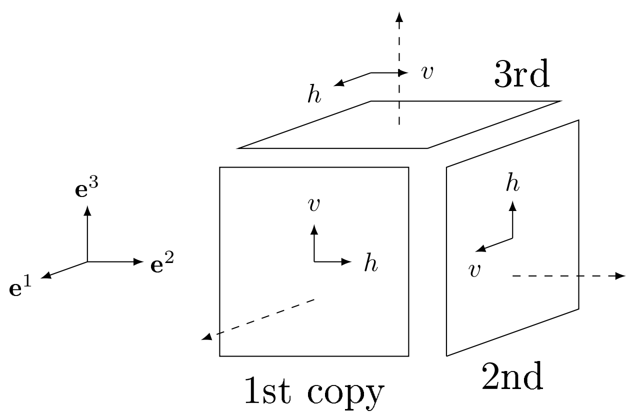


Figure 6.9: Illustration of the first three rules of the three dimensional Robinson subshift. The symbols h and v indicate respectively the horizontal and vertical directions in each of the copies of the subshift X_{adR}.

6.4.1.2 Hierarchical structures

1. **Finite supertiles:**

   In this paragraph, the symbols sw, se, nw, ne mean respectively south west, south east, north west and north east orientations of the corner symbols in the alphabet of the Robinson subshift.

   (a) **Definition by projection on the faces:**
   Any block on the language of this layer whose projection over a plane parallel to $\mathbf{e}^2$ and $\mathbf{e}^3$ (resp $\mathbf{e}^1$ and $\mathbf{e}^3$, resp. $\mathbf{e}^1$ and $\mathbf{e}^2$), considering only the first (resp. second, resp. third) coordinate of the triple, is an order n two-dimensional supertile with orientation t_1 (resp. t_2, resp. t_3) is called a **three-dimensional supertile** with orientation $t \in \{sw, se, nw, ne\}^3$ and order n.

   (b) **Recursive definition:**
   The order $n+1$ three-dimensional supertile with orientation $t \in \{sw, se, nw, ne\}^3$ can be constructed from the order n cubic supertiles as follows. The support of the order $n+1$ cubic supertile is $\mathbb{U}^3_{2^{n+2}-1}$. Figure 6.11 shows positions in the support. On each of the

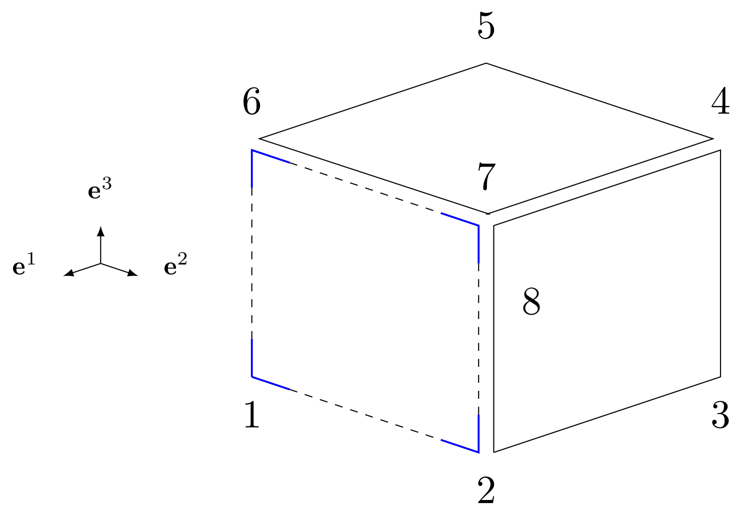


Figure 6.10: Illustration of the coincidence rules of the structure layer: the corners of the cube represent the possibilities of blue corner triples.

translate of $\mathbb{U}^3_{2^{n+1}-1}$ corresponding to these positions, we put the an order n cubic supertile whose orientation is given by the table.

In order to complete the construction we put on position $(2^{n+2}, 2^{n+2}, 2^{n+2})$ a triple of red corners with orientation t. The three planes separating the order n supertiles are filled with three arrows symbols or four arrows symbols induced by the corners. On the three lines of these planes intersecting the center position $(2^{n+2}, 2^{n+2}, 2^{n+2})$ there are triples with a unique corner. Because the corners of the center triple have compatible orientations, these triple are admissible. The reason is that the arrows symbols are orthogonal to the corner.

One can see that in particular, the faces of the cubic supertiles constructed this way verify the recurrence relation used to construct the supertiles of the Robinson subshift.

(c) **Admissibility:**

One can check that the order 1 cubic supertiles are locally admissible patterns. As a consequence, by a recurrence argument, **the three-dimensional supertiles are locally admissible patterns** of this subshift. This means that the supertiles do not admit any forbidden pattern as a sub-pattern).

2. **Three-dimensional cells:**

For all $n \geq 0$, we refer to any subset of $\mathbb{Z}^3$ which is the support of a pattern whose projection over one of the coordinates on the corresponding face (the face which is parallel to the copy of the Robinson subshift) is an order n two dimensional cell as an order n **three-dimensional cell**. The order n **three-dimensional cells** have size $4^n + 1$: this property comes directly from the properties of the two dimensional cells.

3. **Infinite supertiles:**

(a) **Definition:**

Any order n three-dimensional supertile forces the presence of an order $n{+}1$ three-dimensional supertile in the direction of its orientation. This comes from the properties of the rigid ver-

Position of the translate	Orientation of the supertile
(0,0,0)	(ne, ne, ne)
$(2^{n+1}, 0, 0)$	(ne, se, nw)
$(0, 2^{n+1}, 0)$	(nw, ne, sw)
$(2^{n+1}, 2^{n+1}, 0)$	(nw, se, sw)
$(0, 0, 2^{n+1})$	(se, nw, ne)
$(2^{n+1}, 0, 2^{n+1})$	(ne, sw, nw)
$(0, 2^{n+1}, 2^{n+1})$	(sw, nw, se)
$(2^{n+1}, 2^{n+1}, 2^{n+1})$	(sw, sw, sw)

Figure 6.11: Correspondence table for recursive definition of three-dimensional supertiles. The table gives the orientation of order the n cubic supertile superimposed on the $\mathbf{v} + \mathbb{U}^3_{2^{n+2}-1}$, where $\mathbf{v}$ are the entries of the table.

sion of the Robinson subshift X_{adR} listed in Section 3.5.2. For any configuration x of this structure layer, we denote $\sim_x$ the equivalence relation on $\mathbb{Z}^3$ defined by $\mathbf{i} \sim_x \mathbf{j}$ if there is a supertile in x which contains $\mathbf{i}$ and $\mathbf{j}$. This is indeed an equivalence relation, because two supertiles can not intersect but when one of these two supertiles is a sub-pattern of the other. This comes directly from the same fact verified by the Robinson subshift, by projection on the faces of the three-dimensional blocks. An **infinite order** three-dimensional supertile is an infinite pattern over an equivalence class of this relation.

(b) **Types of configurations according to the number of infinite supertiles:** Each configuration is amongst the following types:

(i) A unique infinite order supertile which covers $\mathbb{Z}^3$.

(ii) Two infinite order supertiles separated by a plane. In this case, on the plane, two of the coordinates of the triple are constant and equal to orthogonal three or four arrows symbols (this means that the long arrows of these symbols are orthogonal), whose arrows' directions are in the plane. The last coordinates form a configuration of the Robinson subshift. This is a degenerated non-centered face or internal face of a three-dimensional cell. See Figure 6.12 for an illustration, where the internal faces are colored gray, and the localization of non centered part of one of these faces specified by a red square.

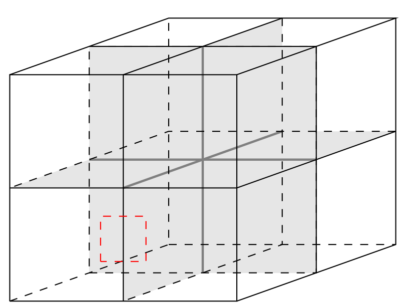


Figure 6.12: Illustration of the internal faces.

(iii) Four infinite order supertiles separated by two orthogonal planes. On the intersection of the two planes, one coordinate is constant and consists in some (red) corner. The other coordinates can be as follows:
   - on the positions of the intersection, there are two arrows symbols with either three or four arrows. This corresponds to a degenerated face superimposed with a configuration of the Robinson with four infinite supertiles.
   - on some position of the intersection, there are two six arrows or two five arrows or two three arrows symbols pointing in the opposite direction of the arms of the corner (and the symbols of the other positions are determined). This corresponds to a degenerated centered edge of a three-dimensional cell, illustrated by point 1 on Figure 6.13
   - all the positions of the intersection have two four arrows symbols or two three arrows symbols pointing to the same direction, orthogonal to the corner. This case corresponds to a degenerated non-centered edge, illustrated by point 2 on Figure 6.13.

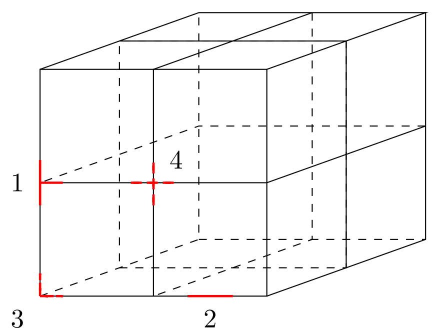


Figure 6.13: Illustration of some locations corresponding to degenerated supertiles.

(iv) Eight infinite order supertiles, separated by three orthogonal planes. The intersection of the three planes contains some corner symbol. Then there are two cases:
   - on this intersection there is a triple of red corners with compatible orientations. This corresponds to a degenerated corner of a cube, illustrated by point 3 on Figure 6.13.
   - there is only one corner, and there are two six arrows symbols or two five arrows symbols pointing in the opposite direction of the arms of the corner. This case corresponds a degenerated centered face of a cube, illustrated by point 4 on Figure 6.13.

(c) **Proof of the exhaustiveness of this classification:**
Let us prove that there is no other possibility.

   i. **Evaluation of the space separating infinite supertiles:**
   First, the set of positions that are outside any infinite supertile does not contain any translate of $\mathbb{U}_2^3$, because it would imply that it contains some triple of blue corners, and there would be an infinite supertile which does not intersects the others (impossible) or intersects non-trivially another infinite supertile (impossible, because they are equivalence classes).

   ii. **Possible supports and combination of them:**
   As for the Robinson subshift, the supports of infinite supertiles are some translates of $(\epsilon_1\mathbb{N}) \times (\epsilon_2\mathbb{N}) \times (\epsilon_3\mathbb{N})$ with $\epsilon_i \in \{-1,1\}$ (1/8 of $\mathbb{Z}^3$), $(\mathbb{Z}) \times (\epsilon_1\mathbb{N}) \times (\epsilon_2\mathbb{N})$, $(\epsilon_1\mathbb{N}) \times$

$(\epsilon_2\mathbb{N}) \times (\mathbb{Z})$, $(\epsilon_1\mathbb{N}) \times (\mathbb{Z}) \times (\epsilon_2\mathbb{N})$ with $\epsilon_i \in \{-1,1\}$ (1/4 of $\mathbb{Z}^3$), some $(\mathbb{Z}) \times (\mathbb{Z}) \times (\epsilon\mathbb{N})$, $(\mathbb{Z}) \times (\epsilon\mathbb{N}) \times (\mathbb{Z})$, $(\epsilon\mathbb{N}) \times (\mathbb{Z}) \times (\mathbb{Z})$ with $\epsilon \in \{-1,1\}$ (half $\mathbb{Z}^3$) or $\mathbb{Z}^3$. Because the set of positions that are not in any infinite supertile does not contain any translate of $\mathbb{U}_2^3$, the possibilities correspond to the type (*i*) to (*iv*) types listed above.

iii. **Reduction of eight infinite supertiles configurations:**
In the case where there are eight supertiles separated by three planes (type (*iv*)), if there are three corners, then they have compatible orientations. If not, then there is one corner. Indeed, if there were no corner, the intersection of the three planes would be composed by three orthogonal arrows symbols. This is impossible (see the coincidence rules). Then the triple has to be of type (1) of the third coincidence rule. Hence, if it was of type (2) for instance, it would mean that, projecting on the copies of the Robinson that do not correspond to corner, the configurations have two infinite supertiles separated by an infinite line, and the plane generated by translating this line is included in the two orthogonal planes generated by translation of the separating cross of the first copy of the Robinson subshift. There are only four infinite supertiles.

iv. **Four infinite supertiles:**
In the case of four infinite supertiles, if one position of the intersection has a corner, then this corner is present on the whole line, and the triples on this line are of type $(2), (3), (4)$ types of the coincidence rule, and to the type (*iii*) of the above description.

v. **Two infinite supertiles:**
When there are two infinite supertiles, by projecting on the copies of the Robinson, we get that two of the copies have two infinite supertiles with a separating line. Moreover, the plane supports a configuration of the Robinson which has a unique infinite supertile, for if it was not the case, there would be two separating planes.

6.4.1.3 Properties of this layer

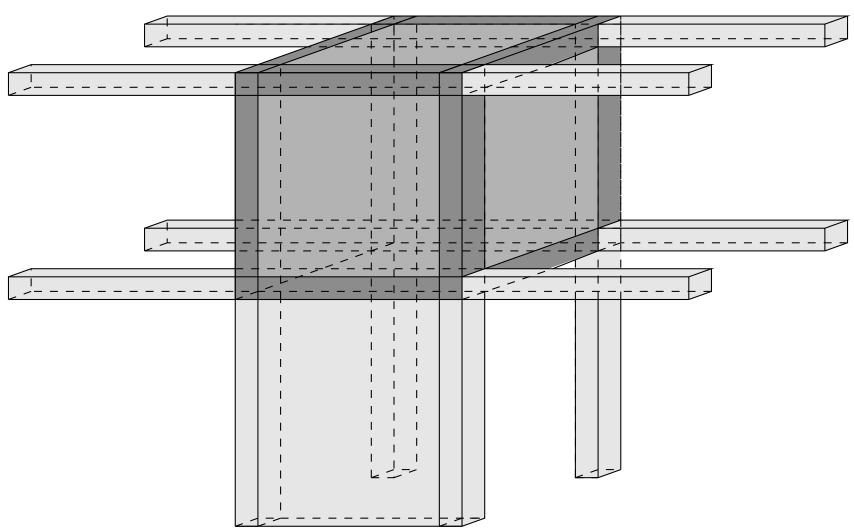


Figure 6.14: Partial representation of the surroundings of a three-dimensional cell - omitting the other ones.

The structure layer has the following properties:

1. **Non-emptiness:**

   A configuration:

   - whose projections on the copies of X_{adR} are all of type (iii) and centered on a red corner
   - and such that the triple of these centers has compatible orientations

   is an element of this subshift, hence it is **not empty**.

2. **Repetition of the supertiles:**

   The order m three-dimensional supertile appear periodically in any order $n \geq m$ three-dimensional supertile with period 2^{m+2}, horizontally and vertically. This comes directly from the similar property of the two-dimensional supertiles. This is also true inside an infinite supertile. Because we use the rigid version of the Robinson subshift presented in Chapter 3, this is also true for the whole configuration, in any configuration of the subshift.

3. **Completion result:**

   Proposition 6.5. *Any n-block in this layer can be completed into an order k cubic supertile, with $k \geq \left\lceil \log_2(n) \right\rceil + 4$. Moreover it can be complete in an order k three-dimensional cell, with*

   $$k \geq \left\lceil \frac{\lfloor \log_2(n) \rfloor}{2} \right\rceil + 2.$$

   We don't write the proof of this proposition, since this is similar to the two-dimensional version of the subshift. Moreover, this version is also minimal:

   Corollary 6.6. *This three-dimensional version of the Robinson subshift is minimal.*

6.4.1.4 Coloration

We use colors in order to simplify the representations of the configurations of this layer. The positions in the edges of the cubes are characterized by having three petal symbols with $0,1$-counter equal to 1 (recall that we call value of the $0,1$-counter the symbols in $\{0,1\}$ on corner symbols of the Robinson subshift), and we represent this by the symbol . The faces positions have exactly two such symbols, and are represented by . The other faces connecting the edges of the cubes are colored with , and are characterized by having a unique petal symbol with $0,1$-counter value equal to 1. See on Figure 6.14 the representation of the surroundings of a three-dimensional cells with these colorations.

6.4.2 Functional areas

In this section, we describe how to draw functional areas on these faces. This means that we attribute local functions to the positions of these faces realizing the global functions of the counters and machine computations. These local functions are the execution of one step of computation (including the incrementation step for the counter), and horizontal and vertical transmission of information.

Moreover, we use an addressing mechanism that allows the selection of sparse sub-areas of this functional areas, so that the global functions do not contribute to entropy dimension.

In this section, each sublayer is presented as the superimposition, on the colored faces, of symbols in a finite set $\mathcal{A}$. As a consequence, the positions in the intersection of two faces are superimposed with a ordered pair of these symbols, and the other positions in the faces with just one of these symbols. In order to keep the descriptions as simple as possible, each sublayer is presented as having alphabet $\mathcal{A}$, while the real alphabet has to

include the elements of $\mathcal{A}^2$, and the real set of rules corresponds to this alphabet. These rules can be deduced easily from the descriptions above.

Moreover, in each sublayer, the non-blank symbols are superimposed on and only on petals of the copy of the Robinson parallel to this face - we will simply refer to these as petals of this face.

We recall that the petals having $0, 1$-counter value equal to 0 are called **transmission petals**, and the other ones are called **support petals**.

6.4.2.1 Orientation in the hierarchy

The purpose of this first sublayer is to give access, to the support petal of each colored face, to the orientation of this petal relatively to the support petal just above in the hierarchy.

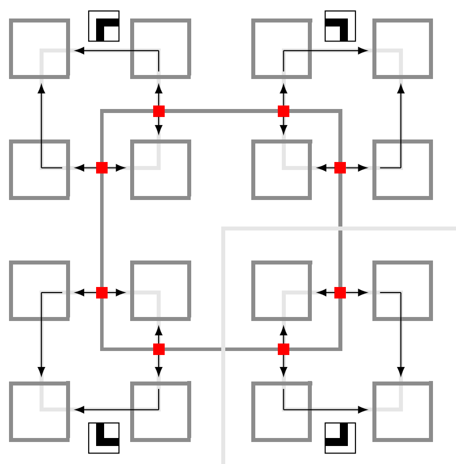


Figure 6.15: Schematic illustration of the orientation rules, showing a support petal and the support petals just under this one in the hierarchy, all of them colored dark gray. The transmission petals connecting them are colored light gray. Transformation positions are colored with a red square. The arrows give the natural interpretation of the propagation direction of the signal transmitting the orientation information.

Symbols:

The symbols are elements of

{□, ┏, ┓, ┛, ┗},

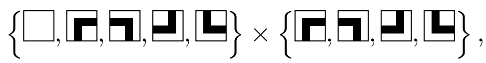
{□, ┏, ┓, ┛, ┗} × {┏, ┓, ┛, ┗},

and a blank symbol.

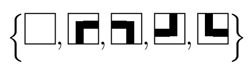
Local rules:

- **Localization:** the non blank symbols are superimposed on and only on positions with petal symbols of the gray faces. The symbol is transmitted through the petals, except on transformation positions, defined just below.

- **Transformation positions:** the **transformation positions** are the positions where the transformation rule occurs (meaning that the signal is tranformed). These positions depend on the (sub)layer. In this sublayer, these are the positions where a support petal intersects a transmission petal just under in the hierarchy. On these positions is superimposed a **ordered pair of symbols**, in

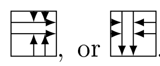


while the other petal positions are superimposed with a **simple symbol**, in



On the transmission positions, the first symbol corresponds to outgoing arrows (this is the direction from which the signal comes, that is to say the support petal), and the second one to incoming arrows (this is the direction where the signal is transmitted, after transformation).

- **Transformation:** on a transformation position, the second bit of the ordered pair is if the six arrows symbol is



These positions correspond to the intersection of an order $n+1$ support petal with the **north west** order n transmission petal just under in the hierarchy. *There are similar rules for the other orientations.*

- **Border rule:** on the border of a gray face, the symbol is blank if not on a transmission position. On a transmission position, the first symbol of the ordered pair is blank.

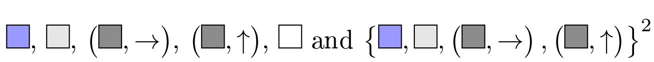
Global behavior:

This layer supports a signal that propagates through the petal hierarchy on the colored faces. This signal is transmitted through the petals except on the intersections of a support petal and a transmission petal just above in the hierarchy. On these positions, the symbol transmitted by the signal is transformed into a symbol representing the orientation of the transmission petal with respect to the support petal.

As a consequence, the support petals just under the transmission petal are colored with this orientation symbol. See the schema on Figure 6.15.

6.4.2.2 Functional areas

Symbols:



Local rules:

- **Localization:** the non blank symbols are superimposed on and only on petal positions of the colored faces.

- the ordered pairs of symbols are superimposed on six arrows symbols positions where the border of a cell intersects the petal just above in the hierarchy (**transformation positions**).

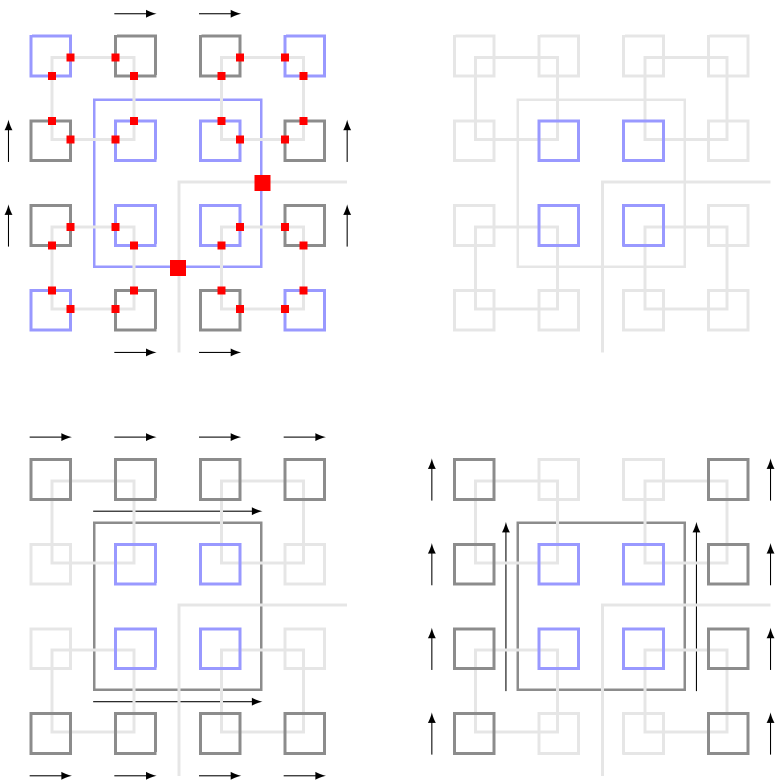


Figure 6.16: Schematic illustration of the transmission rules of the functional areas sublayer. The transformation positions are marked with red squares on the first schema.

- the symbols are transmitted through through the petals except on transformation positions.
- **Transformation through hierarchy**. On the transformation positions, if the first symbol is , then the second one is equal. For the other cases, if the symbol is

  , or ,

  then second color is according to the following rules. The condition on the symbol above corresponds to positions where a support petal intersects the transmission petal just above in the hierarchy and being oriented **north west** with respect to this transmission petal (examples of such positions are represented with large squares on Figure 6.16).

  1. if the orientation symbol is , the second symbol is equal to the first one.
  2. if the orientation symbol is , the second symbol is a function of the first one:
     - if the first symbol of the ordered pair is , then the second symbol is (,→).
     - if the first symbol is (,→), then the second one is equal.
     - if the first symbol is (,↑), then the second symbol is .
  3. if the orientation symbol is , the second color is .

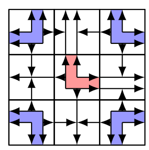
4. if the orientation symbol is , the second symbol is a function of the first one:
   - if the first symbol of the ordered pair is or , then the second symbol is (,→).
   - if the first symbol is (,→), then the second one is .
   - if the first symbol is (,↑), then the second symbol is equal.

For the other orientations of the support petal with respect to the transmission petal just above, the rules are obtained by rotation.

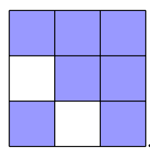
See Figure 6.16 for an illustration of these rules.

- **Border rule:** on the border of a face of a three-dimensional cell, the symbol is blue.
- **Coherence rules:** these rules enable a retroaction of the colors on infinite petals over the order 0 petals that are nearby, ensuring a coherence amongst the hierarchy (they are similar to the the property Q in Mozes' construction).

  1. when near a **corner** of a three-dimensional cell, on a pattern

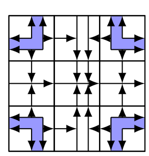
     

     in one of the copies of the Robinson subshift, the corresponding pattern in this sublayer has to be

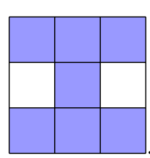
     

     .

     There are similar rules for the other corners. This allows degenerated behaviors which do not happen around finite cells to be avoided. The other rules in this list have the same function. Their analysis is important to understand how it is possible to complete patterns of this subshift into three-dimensional cells.
  2. near an **edge** of a three-dimensional cell, the symbols on the corresponding blue corners on the two faces connected by this edge have to match. Moreover, on the middle of an edge, the two blue corners inside each one of the two faces adjacent to the edge that are on the two sides of the middle, are colored blue. For instance, on the pattern

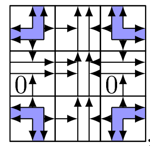
     

     in any of the copies of the Robinson subshift, the pattern in this layer has to be

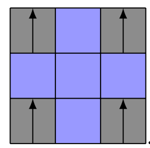
     .

When on the quarter of the edge, the two symbols have to be all (■, $\rightarrow$) or all (■, $\uparrow$), the direction of the arrows corresponding to the orientation of the edge. For instance, on the pattern

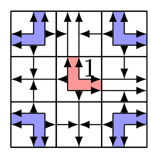


,

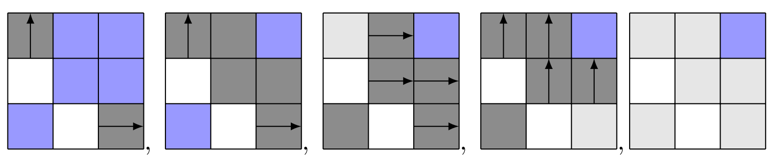
the pattern has to be:


.

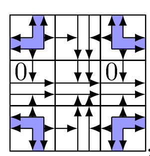
3. Inside a colored face, the degenerated behaviors correspond to the degenerated behaviors of the two-dimensional version of the Robinson subshift. When near a corner of a support petal inside the face, the symbols on the blue corners nearby have to be colored according to the color of the corner. For instance, on



in the copy of the Robinson subshift parallel to the face, the pattern in this layer has to be amongst the following ones:

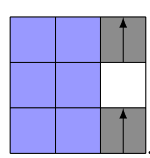


We impose similar restrictions for the other orientations of the corner. On the positions where two petals intersect, the two symbols inside the two-dimensional cell have to be colored with blue, and the two other ones have to be marked with an arrow pointing in the direction parallel to the cell border. For instance, on the pattern



,

the pattern has to be:


.

When near the border of a cell not intersecting another petal, then the only restriction is to have arrows marks outside, in the direction of the border.

Global behavior:

Like the first sublayer presented in Section 6.4.2.1, the global behavior of this sublayer consists in coding a substitution. This coding uses a signal that propagates through the petal hierarchy, on any colored face.

This process is similar as the one used by Robinson [Rob71] in order to create areas supporting the computations of Turing machines in a $\mathbb{Z}^2$-SFT. However, we localize it here on faces of cubes in a three-dimensional SFT.

The result of this process is that order zero two dimensional cell borders of a colored face are colored with a symbol which represent a **function** of the blue corners positions - called **functional positions** - included in the zero order petal just under this petal in the petal hierarchy. These symbols and functions are as follows:

- **blue** if the set of columns and the set of lines in which it is included do not intersect larger order two-dimensional cells. The associated function is to **support a step of computation** (which can be just to transfer the information in the case when the face support a counter), and the corresponding positions are called **computation positions**.
- an **horizontal** (resp. vertical) **arrow** directed to the right (resp. to the top) when the set of columns (resp. lines) containing this petal intersects larger order two-dimensional cells but not the set of lines (resp. columns) containing it. The associated function is to **transfer information** in the direction of the arrow (this information can be trivial in the case when the face support a counter. This means that the symbol transmitted is the blank symbol), and the corresponding positions are called **information transfer positions**.
- when the two sets intersect larger order cells, the petal is colored **light gray**. These positions have no function.

See on Figure 6.17 a schema of the functional positions over the faces of an order 3 three-dimensional cell.

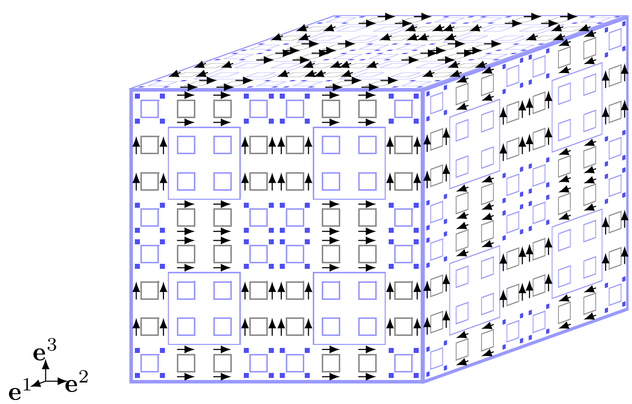


Figure 6.17: Schema of the functional positions on the a face of an order 3 three-dimensional cell. The arrows are oriented according to the fixed orientations of the face.

The aim of the following sections, Section 6.4.2.3, Section 6.4.2.4, and Section 6.4.2.5 is to add symbols on the colored faces in order to specify thin sub-areas of the functional areas. These sub-areas will support the computations of the machines and the linear counters. They are thin enough so that the machines and counters do not contribute to the entropy dimension. This entropy dimension is thus generated by the values of the hierarchical counter.

The first sublayer of this set encodes a counter, called p-counter. It specifies, on each support petal, the arithmetical class of n modulo p, where n is the order of the corresponding two-dimensional cell. The second sublayer, based on this p-counter, gives an address of the columns (resp. lines) amongst the **computation columns** (resp. lines), defined by intersecting computation positions in the face. In the third sublayer, we describe a process which selects a thin subset of these columns (resp. lines) according to the address.

6.4.2.3 The p-counter

We recall that $p = 2^m - 1$ is an integer, fixed at the beginning of the construction.

Symbols:

Elements of $\mathbb{Z}/p\mathbb{Z}$ and of $(\mathbb{Z}/p\mathbb{Z})^2$ and a blank symbol.

Local rules:

- **Localization:**
  - the non blank symbols are superimposed on and only on gray faces, on petals of the copy of the Robinson subshift parallel to this face.
  - The blue petals are superimposed with $\overline{0}$.
  - the ordered pairs are superimposed on **transformation positions** defined as the intersection positions of a support petal and the transmission petal just above in the hierarchy.
  - the other positions have a simple symbol and this symbol is transmitted through these positions.
- **Transformation rule:** on transformation positions, if the first bit is $\overline{i}$, then the second bit is $\overline{i}+\overline{1}$.
- **Coherence rule:** on the edges connecting two faces, the two values of the p-counter are equal. On the corners, the three values are equal.

Global behavior:

Each of the support petals on the colored faces is attached with some element of $\mathbb{Z}/p\mathbb{Z}$. This symbol is transmitted through the petals except on transformation positions. These positions are defined to be the intersections of the support petals with the transmission petal just above in the hierarchy, where the transmission petal has value $\overline{k}+\overline{1}$ and $\overline{k}$ is the value of the support petal. As the blue petals are marked with $\overline{0}$, this imposes that the border petals of the order n two-dimensional cells on the faces are marked with $\overline{n}$.

6.4.2.4 The p-addressing

This sublayer has two subsublayers, one for vertical addresses, and another for horizontal addresses.

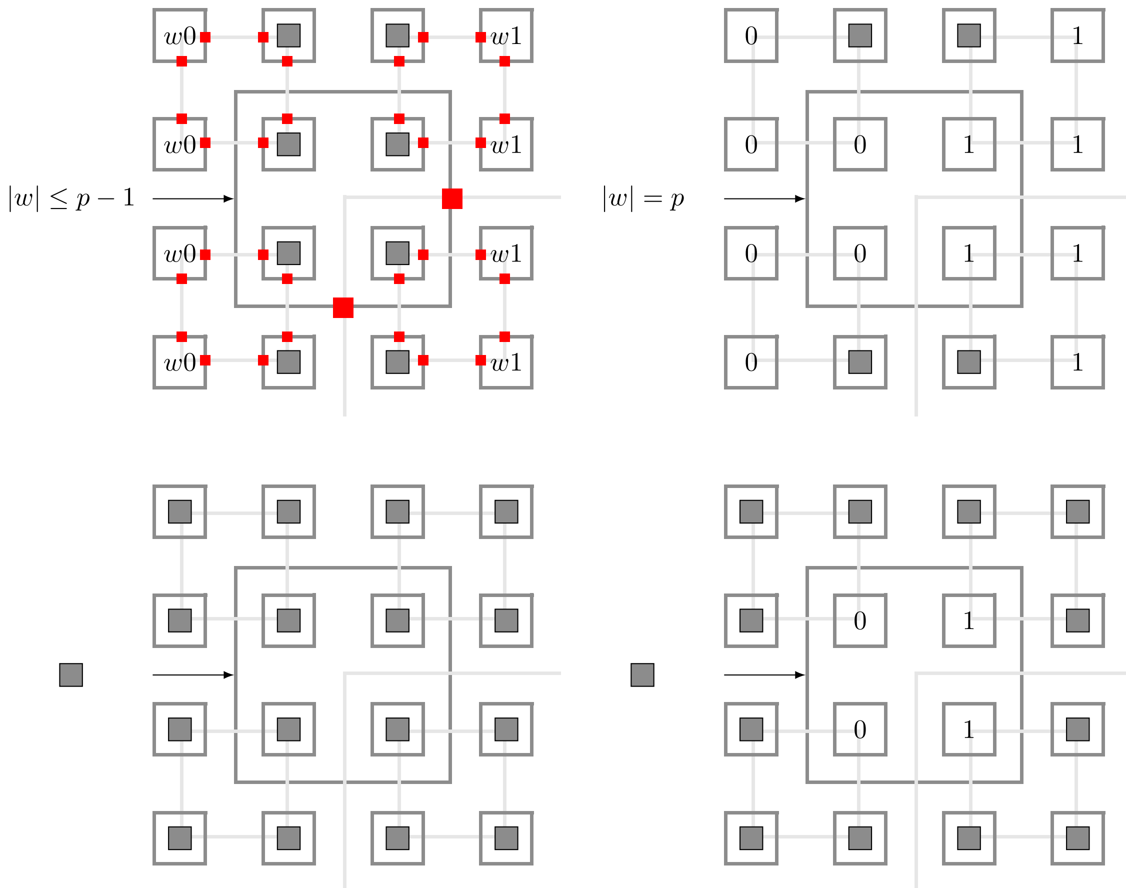


Figure 6.18: Schemata of the transformation rules of the vertical addressing. On the two schemata on the left, the central petal has p-counter value not equal to $\overline{0}$. On the two schemata on the right, this value is $\overline{0}$. The transformation positions are symbolized by red squares. The symbols inside the little petals is the symbol attached to them in this sublayer.

Vertical addressing: ***Symbols:***

The symbols of this subsublayer are the following ones.

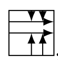
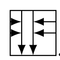
- Length k words on the alphabet $\{0,1\}$, for $k = 1...p$: these symbols are called the **vertical address** of support petals relatively to the next support petal above in the hierarchy.
- The ordered pairs (w, w') such that w and w' are words on alphabet $\{0,1\}$ with respective lengths k and $k+1$ with $k < p$, and $(w, \blacksquare)$ (they represent possible **transformations** of simple symbols when going through a support petal corresponding to a two-dimensional cell having order different from any mp, $m \geq 0$).
- The ordered pairs $(w, 0)$ and $(w, 1)$, where w is a length p word on $\{0,1\}$ (**transformations** of simple symbols when going through a petal corresponding to order mp cells).

- The symbols ■, (■,■), (w,■), (■,0) with w a word on $\{0,1\}$ having length between 1 and p, (■,1),
- A blank symbol.

Local rules:

- **Localization:**
  - the non-blank symbols are located on the petals of the colored faces.
  - the symbols are transmitted through the petals, except on **transformation positions**. These positions are defined as the intersection of a support petal and the transmission petal just above in the hierarchy.
  - the transformation positions are written with an ordered pair, and the other petal positions with a simple symbol.
  - the non-transmission positions in counter $\overline{k}$ support petals with $k \leq p-1$ are superimposed with a length $p-k$ word on $\{0,1\}$ or the symbol ■.
- **Border rule:** on the border of a gray face, all the positions are colored with ■.
- **Transformation through hierarchy:** on the intersection of a petal support with the transmission petal just above in the hierarchy (**transformation positions**):

  if the symbol is

  , or ,

  corresponding to positions symbolized by a large red square on Figure 6.18, then the second symbol is set according to the following rules:

  1. if the orientation symbol is , and the first symbol is some length $k < p$ word w, then the second symbol is $w0$. If the first symbol is some length p word, then the second is 0. If the first symbol is ■, then the second symbol is equal.
  2. if the orientation symbol is , then the second symbol is ■.
  3. if the orientation symbol is , the second symbol is ■ if the counter is not $\overline{p-1}$, and 1 else.
  4. if the orientation symbol is , and the first symbol is some length k word w for $k < p$, then the second symbol is $w0$. If the first symbol is some length p word w, then the second is 0. If the first symbol is ■, then is the second.

  There are similar rules when the Robinson symbol corresponds to another orientation of the support petal with respect to the support petal just above in the hierarchy.

  See Figure 6.18 for an illustration of these rules.

- **Coherence rules:** we impose coherence rules in a similar way as in Section 6.4.2.2. These rules impose that the order 1 petals inside a two-dimensional cell on a gray face are superimposed with ■ if the value of the p-counter of the cell border is not $\overline{0}$ and is a length p word on $\{0,1\}$ if it is $\overline{0}$.

  We don't describe these rules, in order to keep the exposition as simple as possible, since they are similar as in Section 6.4.2.2.

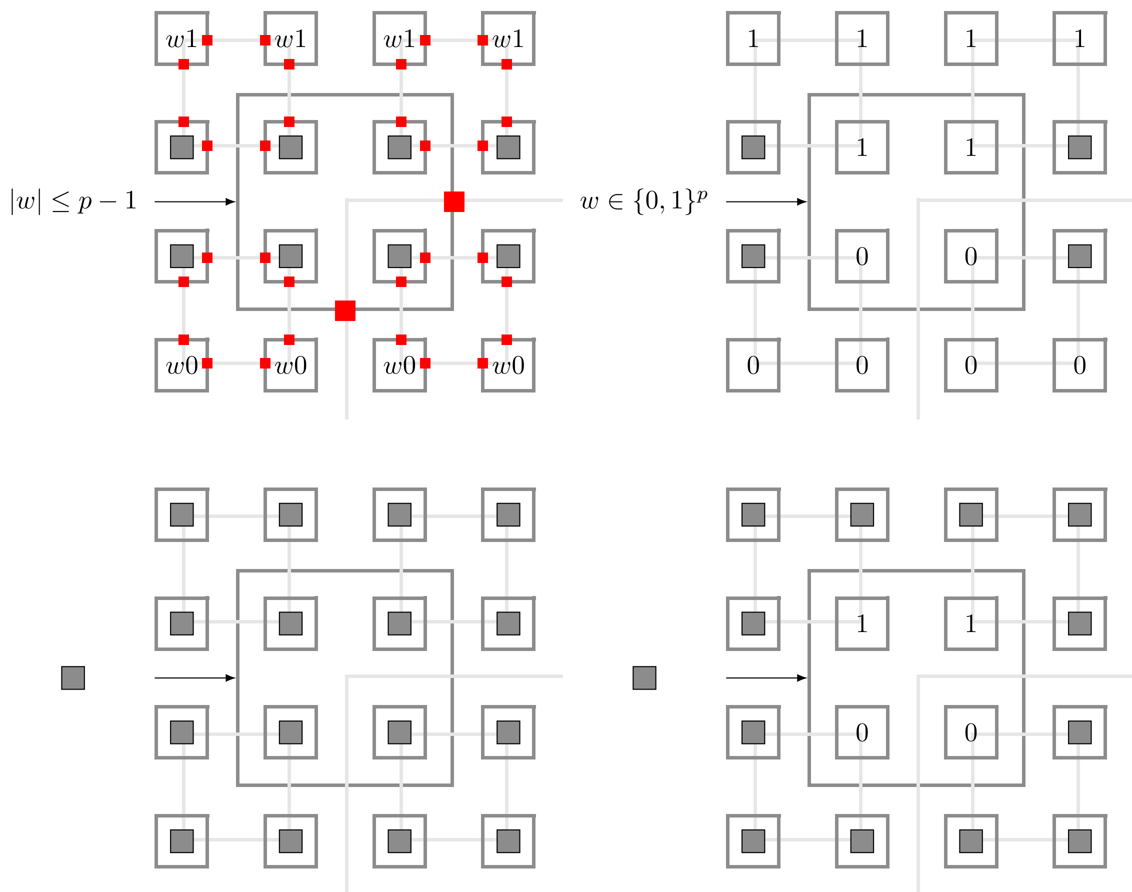


Figure 6.19: Schemata of the transmission rules of the horizontal addressing. The central petals on the two schemata on the left have p-counter value not equal to $\overline{0}$. The one on the schemata on the right have p-counter value equal to $\overline{0}$.

Horizontal addressing: We add another subsublayer with the same symbols and similar rules that are abstracted as on the Figure 6.19.

Global behavior:

In the *vertical addressing* subsublayer, then on each colored face, the petals support the propagation of a signal which is transformed on the intersections of a support petal with the transmission petal just above in the hierarchy. Except on the transformation positions, the petals are colored with a non-empty word in $\{0,1\}$ or the symbol ■. Considering an order n colored face, and three integers k, m and j such that $mp \leq j < (m+1)p \leq kp \leq n$, the support petals that:

1. are the border of an order j two-dimensional cell,

2. are included in an order kp cell,

3. such that the columns in the cells intersecting the order j cell do not intersect any order i cell with $j < i \leq kp$,

are marked with a length $(m+1)p-j$ word on $\{0,1\}$ which codes for the column position of this petal in the order kp cell with respect to the next order $(m+1)p$ cell in the hierarchy.

All the other support petals are colored with $\square$.

The possible length l addresses, with $1 \leq l \leq p$ are all the words in $\{0,1\}^l$, and are arranged in the alphabetic order, 0^l being the address of the leftmost petals and 1^l the address of the rightmost ones.

See an example of addresses arrangement on Figure 6.20.

As a consequence, in an order kp two-dimensional cell, with $k \geq 0$, the columns containing order 0 cells and not intersecting order i cells such that $0 < i \leq kp$, are virtually addressed with a word in $\{0,1\}^{kp}$, obtained as the concatenation of k length p addresses.

The global behavior of the horizontal addressing sublayer is similar, replacing columns by rows.

The aim of the following section is to select a subset of these columns by selecting a subset of the length p addresses.

6.4.2.5 Active functional areas

In this section, we describe how to select, for all the two dimensional cells on the colored faces, a subset of the functional areas of these cells which is sparse enough to not contribute the entropy dimension. This subset is although large enough so that the machines have access to all the data it has to check (namely, the frequency bits, and grouping bits, presented later).

Let Δ be the set of words w in $\{0,1\}^p$ (amongst vertical or horizontal p-addresses) such that there exists $k \in [\![0,p]\!]$ such that $w = 0^k 1^{p-k}$. For instance, when $p = 3$, $\Delta = \{000, 011, 001, 111\}$.

Symbols:

The elements of $\{\square, \blacksquare\}^2$, $\{\square, \blacksquare\}^2 \times \{\square, \blacksquare\}^2$, and a blank symbol.

Local rules:

- **Localization:**
  - the non-blank symbols are located on the petals on gray faces.
  - The **transformation positions** are the positions where a support petal intersects a transmission petal immediately under in the hierarchy. The transformation positions are marked with two ordered pairs, and the other positions with a unique ordered pair.
- **Border rule:** on the border of the faces of the three-dimensional cells, the possible symbols are $\blacksquare$ and $(\blacksquare, \blacksquare)$.
- the symbols are transmitted through petals and intersections of petals that are not transformation positions.
- **Transformation through hierarchy**. On transformation positions we have the following rules:
  - if the counter is not $\overline{0}$, then the two elements of the ordered pair are the same.
  - if the counter is $\overline{0}$, then the ordered pair is changed from the $n+1$ order petal to the n order one as follows:

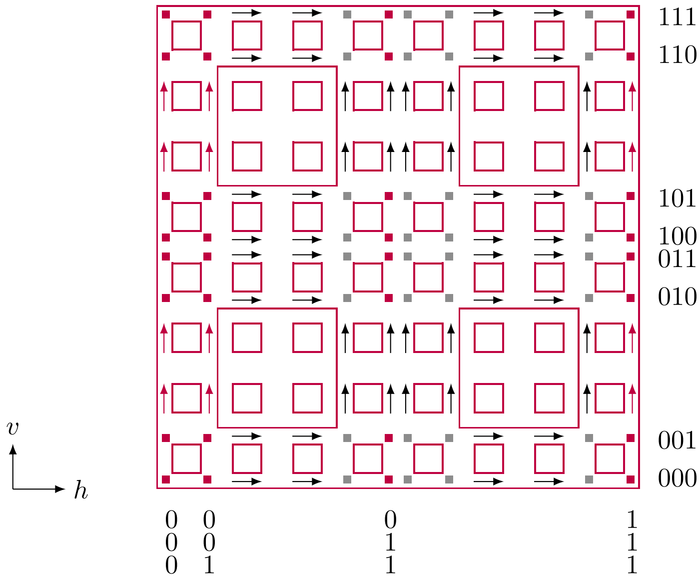


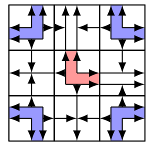
Figure 6.20: Schema of the activation signal in a some $(m+1)p$ order two dimensional cell, when $p = 3$, for the mp order cells contained in it, for any $m \geq 0$. The addresses of the active columns are 000, 001, 011 and 111, and are written under these columns. On the right side are written the addresses of all the lines - not only the active ones.

1. In the following cases, the ordered pairs become (,):
   - $*$ the two addresses are in Δ,
   - $*$ the Robinson symbol is , or , and the orientation symbol is ,
   - $*$ the Robinson symbol is , or , and the orientation symbol is ,
   - $*$ the Robinson symbol is , or , and the orientation symbol is .
   - $*$ the Robinson symbol is , or , and the orientation symbol is
2. if the ordered pair of addresses is in $\Delta \times \Delta^c$, the ordered pair of colors is changed to (,): the first address corresponds to first coordinate, and the second one to the second coordinate.
3. if the ordered pair of addresses is in $\Delta^c \times \Delta$, the ordered pair of colors is changed to (,).
4. in the other cases, the ordered pair of symbols is changed to (,)

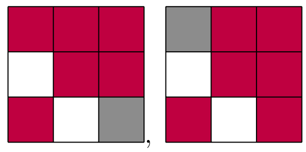
See an illustration on Figure 6.20 when $p = 3$, meaning $m = 2$.

- **Coherence rules:** we consider, in the following list, patterns that intersect three-dimensional cells having order greater or equal to p (on order $\leq p$ we can impose coherence with a finite set of rules).

1. when near a **corner** of a three-dimensional cell, on a pattern

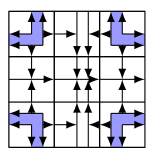


   in one of the copies of the Robinson subshift, the corresponding patterns in this sublayer has to be (respectively first and second coordinates):

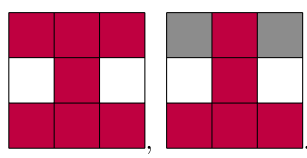


   with similar rules for the other corners.

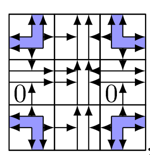
2. near an **edge**, for the corresponding blue corners on the two faces, the symbols have to be equal. On the middle of an edge, the blue corners are colored with in the coordinate corresponding to the direction of the edge. The other coordinates are such that when the p-counter value on the edge is $\overline{0}$, the bottom blue corners in the case of a vertical edge, and the left ones in the case of a horizontal edge, are colored with purple, and the other two with gray. When the p-counter value on the edge is not $\overline{0}$, then all the blue corners are colored purple. For instance, on the following pattern with counter $\overline{0}$ on the edge



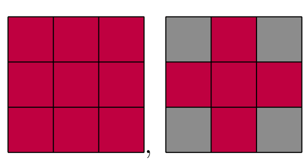
   in any of the copies of the Robinson subshift, the patterns in this layer has to be



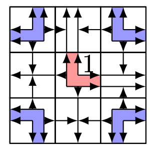
   When on the quarter of the edge, the blue corners are colored with in the coordinate corresponding to the direction of the edge, and the other coordinate is . For instance, on the pattern



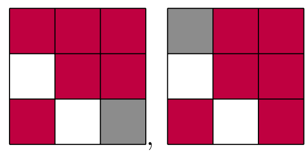
   the patterns has to be:



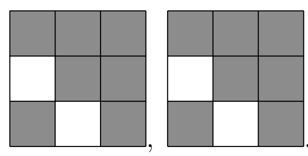
3. Inside a face of a three-dimensional cell, the degenerated behaviors correspond to the degenerated behaviors of the $2d$ version of the Robinson subshift. We consider only two-dimensional cells having order greater or equal to p. When near a corner of a support petal inside the area, the symbols on the blue corners nearby have to be colored according to the color of the corner. On



in the copy of the Robinson subshift parallel to the face, the patterns (respectively first and second coordinate) in this layer have to be

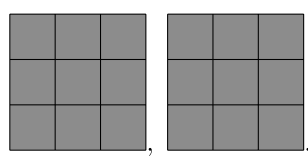


or



We have similar restrictions for the other orientations of the corner.

On the intersection positions of two petals, the possibilities are the same as on the border of the three-dimensional cell, except is added the possibility that the two patterns of the ordered pair are all gray. On the same example:



When near the border of a cell not intersecting another petal, or a line with three arrows symbols connecting corners, the symbols corresponding to the direction of the line are purple on the left and gray on the right when it is vertical, and purple on the bottom and gray on the top when it is horizontal. The symbols of the other coordinates on both sides have to match.

4. **Coherence with the functional areas:** the symbol (,) in this sublayer can be superimposed only on in the functional areas sublayer. A symbol with a vertical arrow in the functional areas sublayer can be superimposed only with else (,) or (,) in this sublayer. A symbol with a horizontal (resp. vertical) arrow is superimposed with else (,) or (,) (resp. (,)) in this sublayer. When the symbol in the functional areas sublayer is , then in this layer it is (,).

Global behavior:

In this sublayer, on each of the colored faces, the order 0 cells borders are colored with else or . The ones colored with are exactly those that:

1. are contained in an order kp, $k \geq 0$, two-dimensional cell,
2. contained in columns of this cell that do not intersect any intermediate order cell,
3. and have virtual address written $w_1..w_k$, where the words w_i are length p words on $\{0, 1\}$ in the set
$$\Delta = \{0^j 1^{p-j} : 0 \leq j \leq p\}.$$

Such columns are called **active columns** when the order 0 cells border they intersect are colored $\textcolor{red}{\blacksquare}$.

These petals are colored with a second color, with similar rules, replacing columns by lines. The lines colored $\textcolor{red}{\blacksquare}$ are called **active lines**.

Lemma 6.7. *1. On any order n colored face, the number of active columns (resp. active lines) that are not in an order $jq < n$ two-dimensional cell is $2.2^r.(p+1)^q$, where $n = qp + r$, with $r < p$.*

2. The number of active columns (resp. active lines) in an order $2k + 2$ supertile (centered on an order k two-dimensional cell) on a gray face is less than $2^{p+1}(p + 1)^q$, where $k = qp + r$, with $r < p$.

Proof. 1. (a) The factor 2^r corresponds to the order qp cells marked with $\textcolor{gray}{\blacksquare}$ in the addresses sublayer: there are 2^r columns (resp. lines) of them.

(b) Then the process of this layer selects $p+1$ addresses of order $p(q-1)$ cells relatively to the pq cells. For these cells, it selects p addresses of order $p(q-2)$ cells relatively to these order $p(q-1)$ cells, etc. Hence the factor $(p + 1)^q$ in the formula.

(c) The factor 2 comes from the fact that there are two active columns for all order 0 cells.

2. There are two possible cases for the color of the maximal order petal in the supertile: $\textcolor{gray}{\blacksquare}$ or $\textcolor{red}{\blacksquare}$. In the first case, there are no active columns or lines. In the second one the argument is similar to the first point.

□

Corollary 6.8. *The number of active functional columns in a cell of order qp, for q integer, is 2^{mq}.*

Lemma 6.9. *For all $k < n$, on the left of the first k order cell in an order n cell (from left to right), the column is active. Moreover, denote p_k the number such that this column is the p_kth column amongst the active ones from left to right. There is an algorithm that computes p_k given as input k.*

Proof. 1. **If k is between pq and $pq+r$,** all the order pq cells are selected by the process (colored purple). We pick the one which is just on the right of the leftmost k order cell. The order $p(q-1)$ cell whose address is 1...1 relatively to this order pq cell is selected. We repeat this argument, considering a sequence of $p(q-k)$ order cells, the last one being the order 1 cell just on the right of the k order cell. The column on the right of the order 1 cell is an active column which is just on the left of the k order cell.

2. **Other values of k:**

For the others k, the argument is similar. Consider some q' such that $q'p \leq k < (q' + 1)p$. Pick an order $(q' + 1)p$ order cell which is selected by the process, and then the order $q'p$ order cell whose address relatively to the first one represents the number $2^{k-(q'p+1)-2} - 1$. Then choose successively the corresponding order $(q' - 1)p$, ... ,0 rightmost cells. The column on the right of the last one is on the left of the picked order k cell.

□

Remark 40. *The reason of the choice of p this way is the factor $(p+1)^q$ in the first point of Lemma 6.7: we need that this number is a power of two, so that the linear counters have as period a Fermat number.*

6.4.2.6 Propagation of the border information inside the colored faces

The point of this section is to give access to all the functional positions in a face of a three-dimensional cell (specified by $\square$) to the value of the p-counter of its border, and a border bit that specifies if a functional position is proper to the face as a two-dimensional cell.

Symbols:

The elements of $\mathbb{Z}/p\mathbb{Z} \times \{0,1\}$ and a blank symbol.

Local rules:

- non-blank symbols are superimposed on the faces of the three-dimensional cells. On these faces, they are superimposed on and only on positions having a Robinson symbol (for the copy which is parallel to the face) which is not a two-dimensional cell border symbol.
- a symbol propagates to any position in the neighborhood positions if this position is not in the border of a two-dimensional cell.
- when on a position near the border of a two-dimensional cell from inside (the inside and outside are characterized by the position of the arrows: left or right for vertical arrows, and top or bottom for horizontal ones), then the symbol is equal to the value of the p-counter on this border. The bit in $\{0,1\}$ (called **border bit**) is 1 when the border is colored with $\blacksquare$ and 1 if colored with $\square$ or $\square$.

Global behavior:

All the functional positions of a face are colored with the value of the p-counter of its border. The ones that are proper to this face as a two-dimensional cell are colored with the border bit 1. The other ones with 0. This is done using the diffusion of a signal on the face. The propagation is stopped by the cells walls. The information transported by these signals is determined on the border of the diffusion areas.

6.4.3 Frequency bits layer

The frequency bits will help to drive a process that, through signaling in the hierarchy, will generate the entropy dimension of the subshift.

Symbols:

The elements of $(\{0,1\} \cup \{\square\})^3$.

Local rules:

- **First coordinate:** non blank symbols for the first coordinate of the ordered pair are superimposed on and only on np cell border positions in the copy of the Robinson subshift parallel to $\mathbf{e}^2$ and $\mathbf{e}^3$.
- **Second coordinate:** non blank symbols for the second coordinate of the ordered pair are superimposed on and only on order np cell border, $n \geq 0$ petal positions in the copy of the Robinson subshift parallel to $\mathbf{e}^3$ and $\mathbf{e}^1$.

- **Third coordinate:** non blank symbols for the third coordinate of the ordered pair are superimposed on and only on order np cell border positions in the copy of the Robinson subshift parallel to $\mathbf{e}^1$ and $\mathbf{e}^2$.
- This means that on a position in $\mathbb{Z}^3$, the number of non-blank coordinates correspond to the color of this position in the structure layer: ■ means three, ▩ means two, ▢ one and □ zero.
- **Synchronization rule:** on positions colored with ■, the three symbols are equal.

Global behavior:

For all $n \geq 0$, the three dimensional cells having order np are attached with the same bit called **frequency bit** and denoted $f_{bit}(n)$. Part of the work of the machine implemented in the following will be to impose that $f_{bit}(n) = a_n$ for all n.

6.4.4 Grouping bits

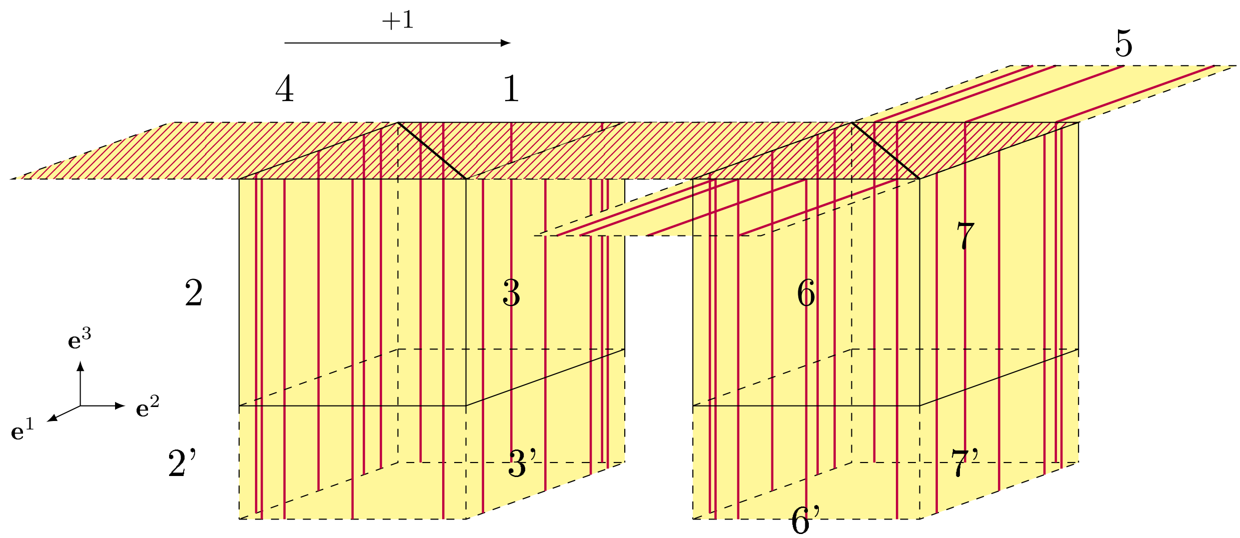
This layer supports bits, called **grouping bits**, attached to the two-dimensional cells of the hierarchy in each copies of the Robinson subshift, having order qp for some q. They verify similar rules as the frequency bits, except that the Turing machines check that the 2^kth (this corresponds to some sparse subset of the counter $\overline{0}$ two-dimensional cells) bit of this sequence is 1 for all k, and the other ones are 0.

The grouping bits serve to make a group of random bits, described later, into values of the hierarchical counter.

6.4.5 Linear counter layer

The construction model of M. Hochman and T. Meyerovitch [HM10] implies degenerated behaviors of the Turing machines. For this reason, in order to preserve the minimality property, we use a counter which alternates all the possible behaviors of these machines. We describe this counter here. We attribute to each face a type, as follows. To each type correspond a specific alphabet and specific rules.

- The top face of three-dimensional cells according to direction $\mathbf{e}^3$ (that we call **type** 1 **face**) supports the following functions:
  - incrementation of the counter value in direction $\mathbf{e}^2$. This value is a word written on the intersection of active columns and the bottom line of the face (Section 6.4.5.2).
  - Extraction of some information (Section 6.4.5.3 and Section 6.4.5.4) from this value.
- the other faces serve for transporting some informations (Section 6.4.5.5) to the machine face. This is the bottom face of the three-dimensional cells according to direction $\mathbf{e}^3$. I serve also for the synchronization of the counter values on three-dimensional cells having the same order and adjacent in directions $\mathbf{e}^1$ and $\mathbf{e}^3$. The definition of these face is as follows:
  - The type 4 (resp. 5) faces are defined as connecting type 1 faces in direction $\mathbf{e}^2$ (resp. $\mathbf{e}^1$).
  - The type $2, 3$ faces are defined as bottom (resp. top) faces of three-dimensional cells according to direction $\mathbf{e}^2$.
  - The type $6, 7$ faces are defined as top (resp. bottom) faces of three-dimensional cells according to direction $\mathbf{e}^1$.
  - The type $2', 3', 6', 7'$ faces are defined as connecting two type $2, 3, 6, 7$ faces in direction $\mathbf{e}^3$.



$$1 : \left(\mathcal{A}_c \times \{0,1\} \times \{\blacksquare, \square\} \bigcup \mathcal{A}_c \times \{\blacksquare, \square\} \bigcup \{\blacksquare, \square\} \bigcup \{0,1\} \times \{\blacksquare, \square\}\right) \times \left(\mathcal{A}_c \bigcup \{\square\}\right)$$

$$4 : \mathcal{A}_c \times \{\blacksquare, \square\} \bigcup \{\blacksquare, \square\}$$

$$2, 2' : \mathcal{A} \times \mathcal{Q} \times \mathcal{D} \times \{\texttt{on}, \texttt{off}\}$$

$$3, 3' : \{\leftarrow, \rightarrow\} \times \{\texttt{on}, \texttt{off}\}$$

$$5 : \mathcal{A}_c$$

$$6, 6', 7, 7' : \mathcal{Q} \times \{\texttt{on}, \texttt{off}\}$$

Figure 6.21: Localization of the linear counter area (colored yellow). In purple, the localization of non-blank symbols in these areas. The demultiplexing line is represented as a thick dark line. The type of information supported by each face in this layer is listed under the picture.

See Figure 6.21 for an illustration.

We describe the global behavior at the end of these sections, and the global behavior for the whole layer is written in Section 6.4.5.6.

In Section 6.4.5.1 we give some notations used in the construction of this layer.

6.4.5.1 Alphabet of the linear counter

Let $l \geq 1$ be some integer, and $\mathcal{A}$, $\mathcal{Q}$ and $\mathcal{D}$ some finite alphabets such that $|\mathcal{A}| = |\mathcal{Q}| = 2^{2^l}$, and $|\mathcal{D}| = 2^{4.2^l - 2}$. Denote $\mathcal{A}_c$ the alphabet $\mathcal{A} \times \mathcal{Q}^3 \times \mathcal{D} \times \{\leftarrow, \rightarrow\} \times \{\texttt{on}, \texttt{off}\}^2$.

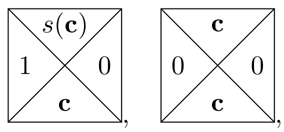
The alphabet $\mathcal{A}$ will correspond to the alphabet of the working tape of the Turing machine after completing it such that it has cardinality equal to 2^{2^l} (this is possible by adding letters that interact trivially with the machine heads, and taking l great enough). The alphabet $\mathcal{Q}$ will correspond to the set of states of the machine (after similar completion). The arrows will give the direction of the propagation of the error signal. The elements of the set $\{\texttt{on}, \texttt{off}\}^2$ are a coefficients telling which ones of the lines and columns are active for computation (which has an influence on how each computation positions work), and the alphabet $\mathcal{D}$ is an artifact so that the cardinality of $\mathcal{A}_c$ is $2^{2^{l+3}}$, in order for the counter to have a period equal to a Fermat number.

Let us fix s some cyclic permutation of the set $\mathcal{A}_c$, and $\mathbf{c}_{\mathtt{max}}$ some element of $\mathcal{A}_c$.

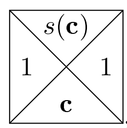
6.4.5.2 Incrementation

Symbols:

The elements of $(\mathcal{A}_c \times \{0,1\}) \times \{\blacksquare, \square\}$. The elements of $(\mathcal{A}_c \times \{0,1\})$ are thought as the following tiles. The first symbol represents the south symbol in the tile, and the second one representing the west symbol in the tile:



for $\mathbf{c} \neq \mathbf{c}_{\mathtt{max}}$, and



for $\mathbf{c} = \mathbf{c}_{\mathtt{max}}$.

The second is called the **freezing symbol**. The other symbols are the elements of the following sets: $\mathcal{A}_c \times \{\blacksquare, \square\}$, $\{\blacksquare, \square\}$,$\mathcal{A}_c$, $\{0,1\} \times \{\blacksquare, \square\}$

Local rules:

- **Localization rules:**
  - in this sublayer, the non-blank symbols are superimposed on type 1 faces such that the value of the p-counter is $\overline{0}$ and the border bit is 1 (we recall that any position in the face is colored with this information).
  - The elements $\mathcal{A}_c \times \{\blacksquare, \square\}$, appear on the active columns, except the bottom line (according to direction $\mathbf{e}^2$).
  - The elements of $\{\blacksquare, \square\}$, appear on all the positions of the face except the border and the bottom line of the functional area.
  - The elements of $\mathcal{A}_c \times \{0,1\} \times \{\blacksquare, \square\}$ appear on computation positions of the bottom line.
  - The elements of $\{0,1\} \times \{\blacksquare, \square\}$ appear on the other positions of the bottom line. See an illustration on Figure 6.22

  A **value** of the counter is a possible sequence of symbols of the alphabet $\mathcal{A}_c$ that can appear on a row, on the positions where this row intersects an active column.

- **Freezing signal:**
  - **On the bottom line:**

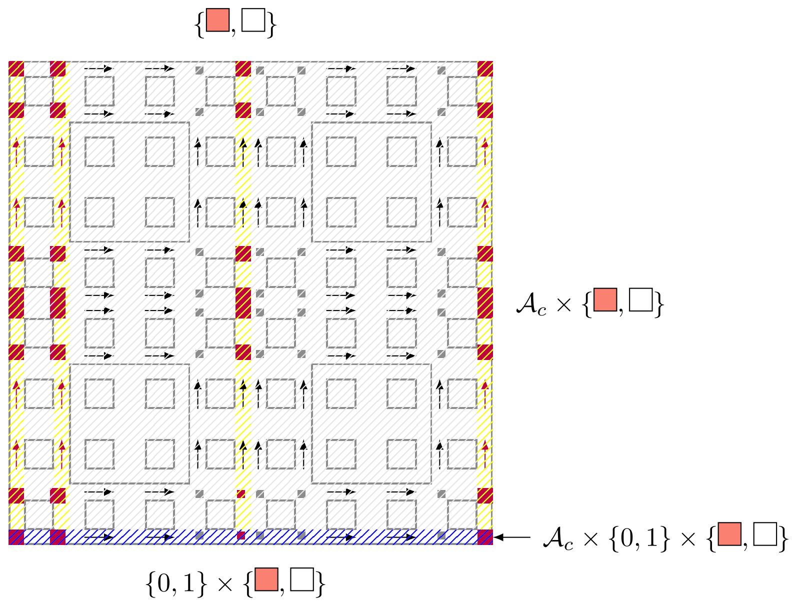


Figure 6.22: Localization of the linear counter symbols on face 1.

1. on the leftmost bottommost position, the color is ■ if and only if the south symbol in the tile is $\mathbf{c}_{\mathtt{max}}$.
2. this color propagates towards the right while the south symbol in the tile is $\mathbf{c}_{\mathtt{max}}$. When this is not true, the color becomes white.
3. the white symbol propagates to the left.

– **Other positions:**

1. the color part of the symbol propagates on the gray area on Figure 6.22.
2. on the bottom right position of this area (on the top of the position on the bottom right corner of the face), the color is white if the color of the position on bottom is white. When this color is salmon (meaning the value of the counter is maximal), then, if the top right position of the adjacent face 4 is colored salmon, then the considered position is colored white. If not, then it is colored salmon. See Figure 6.23 for an illustration of possible freezing symbol configuration on face 1.

- **Incrementation of the counter:**

  On the bottom line of the area:

  – on the leftmost position of the bottom line, if the freezing signal of the top left position of the face 4 under is white, then the east symbol in the tile is 1 (meaning that the counter value is incremented in the line). Else it is 0 (meaning this is not incremented).

  – on a computation position of this line, the part in $\{0, 1\}$ of symbol of the position on the right is the east symbol of the tile. The symbol on the left position is the west symbol of the tile. The symbol on the top position is the north symbol of the tile, and the symbol of the position in type 4 face under is equal to the south symbol of the tile.

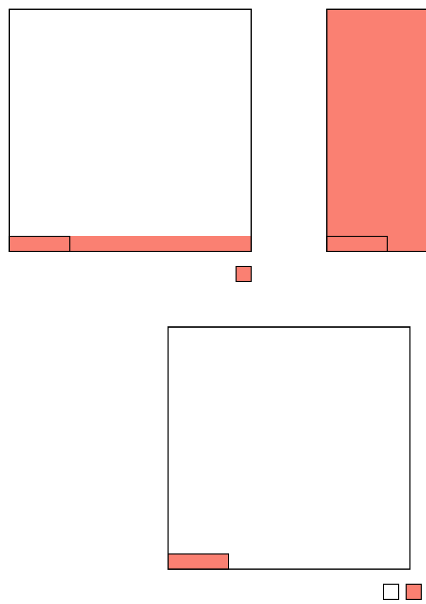


Figure 6.23: Possible freezing symbol configurations on face 1.

  – between two computation positions, the symbol in $\{0,1\}$ is transported.

- **Transfer of state and letter:** in the active columns, while not in the bottom row, the coefficient is transported.

Global behavior:

On type 1 faces having order qp for some $q \geq 0$, the counter value is incremented on the bottom line using an adding machine coded with local rules, except when the freezing signal on face 4 under is ■. Then the value is transmitted through the face 4 above to the face 1 of the adjacent three-dimensional cell having the same order in direction $\mathbf{e}^2$. As a consequence, the counter value is incremented cyclically in direction $\mathbf{e}^2$ each time going through a three-dimensional cell. The freezing signal happens each time that the counter reaches its maximal value and stops the incrementation for one step, since during this step, the freezing signal is changed into □.

6.4.5.3 Demultiplexing lines

Symbols:

⧄ and □.

Local rules:

- **Localization:** the non-blank symbols are superimposed on type 1 faces.

- when not on a position with a corner in the copy of the Robinson subshift parallel to $\mathbf{e}^1$ and $\mathbf{e}^2$, the symbol ⧄ propagates in directions $\pm(\mathbf{e}^1 + \mathbf{e}^2)$.

- when on the north east (resp. south west) corner of the face, the position has symbol $\boxslash$. It forces the same symbol in the direction $-(\mathbf{e}^1 + \mathbf{e}^2)$ (resp. $\mathbf{e}^1 + \mathbf{e}^2$).
- the symbol $\boxslash$ can not cross the border of a two-dimensional cell inside the face, except through south west and north east corners.

Global behavior:

On the top face, according to direction $\mathbf{e}^3$, of every three-dimensional cells, the diagonal of this face joining the south west corner and the north east one is colored with the symbol $\boxslash$ (we recall that each face has specific orientation). See an illustration on Figure 6.21.

6.4.5.4 Demultiplexing

Symbols:

The symbols of this sublayer are the elements of $\mathcal{A}_c$ and a blank symbol.

Local rules:

- **Localization rule:** The non-blank symbols are superimposed on the active lines of type 1 faces where that the value of the p-counter is $\overline{0}$ and the border bit is 1.
- **Transmission rule:** the symbols are transmitted through the columns of the face.
- **Demultiplexing rule:** on symbol $\boxslash$ in the demultiplexing lines sublayer, the symbol in the present sublayer is equal to the symbol in the incrementation sublayer, except on the bottom line. In this case, the north symbol of the tile is copied.

Global behavior:

As a consequence of the local rules, the value of the counter is copied on the diagonal of type 1 faces having order qp and transmitted on the active lines of the faces, so that it can be transmitted to type $5, 6, 6', 7, 7'$ faces.

6.4.5.5 Information transfers

Symbols:

The symbols of this sublayer are the elements of the following sets: $\mathcal{A}_c \times \{\blacksquare, \square\}$, $\{\blacksquare, \square\}$, $\mathcal{A}_c$, $\mathcal{A} \times \mathcal{Q} \times \mathcal{D} \times \{\texttt{on}, \texttt{off}\}$, $\{\leftarrow, \rightarrow\} \times \{\texttt{on}, \texttt{off}\}$, $\mathcal{Q} \times \{\texttt{on}, \texttt{off}\}$, and a blank symbol.

Local rules:

- **Localization rules:** The non-blank symbols are superimposed on the positions of faces such that the value of the p-counter is $\overline{0}$ and the border bit is 1. According to the face type, the possible non-blank symbols and the location on the face are as follows:
  - type $2, 2'$: elements of $\mathcal{A} \times \mathcal{Q} \times \mathcal{D} \times \{\texttt{on}, \texttt{off}\}$, appearing on active lines.
  - type $3, 3'$: elements of $\{\leftarrow, \rightarrow\} \times \{\texttt{on}, \texttt{off}\}$, appearing on active lines.
  - type 5: elements of $\mathcal{A}_c$, appearing on active lines.
  - type $6, 6', 7, 7'$: elements of $\mathcal{Q} \times \{\texttt{on}, \texttt{off}\}$, appearing on active columns.

- type 4: elements of $\mathcal{A}_c \times \{\blacksquare, \square\}$, appearing on active columns, and elements of $\{\blacksquare, \square\}$, appearing on the other positions of the face.

- **Information transfer rules:**
  - On type $2, 2', 3, 3', 5$ faces, the symbols are transmitted through the rows.
  - On type $6, 6', 7, 7'$, the symbols are transmitted through columns.
  - On type 4 faces, the symbols in $\mathcal{A}_c \times \{\blacksquare, \square\}$ are transmitted through columns, and the ones in $\{\blacksquare, \square\}$ are transmitted through all the face, and the part of the symbols in $\mathcal{A}_c \times \{\blacksquare, \square\}$ in the set $\{\blacksquare, \square\}$ is equal to the symbols outside the column.

- **Connection rules:**
  - Across the line connecting face $2, 3, 6, 7$ to face $2', 3', 6', 7'$, the symbols are equal.
  - Across the line connecting face 4 to face 1, on the active columns, the symbols in $\mathcal{A}_c$ on face 4 is equal to the north symbol of the tile on face 1.
  - Across the line connecting faces 2 and $2'$ (resp. 3 and $3'$,6 and $6'$,7 and $7'$,5) to face 1, if the symbol on the north of the tile on face 1 in the incrementation sublayer (resp. incrementation, demultiplexing, demultiplexing, and demultiplexing layers) is written

    $$w = (a, q^1, q^2, q^3, d, f, o^1, o^2) \in \mathcal{A} \times \mathcal{Q}^3 \times \mathcal{D} \times \{\rightarrow, \leftarrow\} \times \{\texttt{on}, \texttt{off}\}^2,$$

    then the corresponding symbol on face 2 is equal to (a, q^1, d, o^1) (resp. (f, o^1), (q^2, o^2), (q^3, o^2), w).

Global behavior:

In this sublayer, the information of the counter value supported by face 1 having order qp, $q \geq 0$, after its incrementation in the first row of this face, is split. The parts are transmitted to faces $2, 3, 6, 7$ where they will be used by the computing machines. Face 4 transfers information for the incrementation mechanism, and faces $2', 3', 6', 7', 5$ allow the synchronization of the counter of adjacent three-dimensional cells having the same order in directions $\mathbf{e}^1$ and $\mathbf{e}^2$.

6.4.5.6 Global behavior

On the top faces of the order qp three-dimensional cells, $q \geq 0$, the counter value represents a length mq word on alphabet $\mathcal{A}_c$. It is incremented when going from a three-dimensional cell to the adjacent one in direction $\mathbf{e}^2$, except once in a cycle, when the counter reaches its maximal value. This time, the value is incremented in two steps. Since the number of active columns is 2^{mq} (Corollary 6.8), the period of the counter (meaning the number of time one has to jump from a three-dimensional cell to the adjacent one in direction $\mathbf{e}^2$ to see the same value) is equal to

$$(2^{8.2^l})^{2^{mq}} + 1 = 2^{2^{l+3+mq}} + 1.$$

Since m and l are fixed, all these numbers, for $q \geq 0$, are two by two different. The important fact about them is that there are all coprime (Lemma 6.14).

Moreover, for two adjacent cells in directions $\mathbf{e}^1$ or $\mathbf{e}^3$, their counter values are equal.

6.4.6 Machines layer

In this section, we present the implementation of Turing machines.

The support of this layer is the bottom face of three-dimensional cells having order qp for some $q \geq 0$, according to direction $\mathbf{e}^3$.

In order to preserve minimality, simulate each possible degenerated behavior of the machines, we use an adaptation of the Turing machine model as follows, slighlty different from the one presented in Chapter 4. The bottom line of the face is initialized with symbols in $\mathcal{A} \times \mathcal{Q}$ (we allow multiple heads). The sides of the face are "initialized" with elements of $\mathcal{Q}$ (we allow machine heads to enter at each step on the two sides). As usual in this type of constructions, the tape is not connected. Between two computation positions, the information is transported. In our model, each computation position takes as input up to four symbols coming from bottom and the sides, and outputs up to two symbols to the top and sides. Moreover, we add special states to the definition of Turing machine, in order to manage the presence of multiple machine heads. We describe this model in Section 6.4.6.2, and then show how to implement it with local rules in Section 6.4.6.3.

In Section 6.4.6.1 we describe signals which activate or deactivate lines and columns of the computation areas. These lines and columns are used by the machine if and only if they are active. These signals are determined by the value of the linear counter.

The machine has to take into account only computations starting on well initialized tape and no machine head entering during computation. For this purpose, we use error signals, described in Section 6.4.7.

6.4.6.1 Computation-active lines and columns

In this section we describe the first sublayer.

Symbols:

Elements of $\{\texttt{on}, \texttt{off}\}^2$, of $\{\texttt{on}, \texttt{off}\}$ and a blank symbol.

Local rules:

- **Localization rules:**
  - the non-blank symbols are superimposed on active lines and active columns positions on a the bottom face according to direction $\mathbf{e}^3$, with p-counter equal to $\overline{0}$ and border bit equal to 1.
  - the ordered pairs are superimposed on intersections of an active line and an active column, the simple symbols are superimposed on the other positions.
- **Transmission rule:** the symbol is transmitted along lines/columns. On the intersections the second symbol is equal to the symbol on the column. The first one is equal to the symbol on the line.
- **Connection rule:** Across the line connecting type $6, 7$ (resp. $2, 3$) face and the machine face, and on positions where the bottom line intersects with active columns, the symbol in $\{\texttt{on}, \texttt{off}\}$ is equal to the first (resp. second) element of the ordered pair in $\{\texttt{on}, \texttt{off}\}$ in this layer.

Global behavior:

On the machine face of any order qp three-dimensional cell, the active columns and lines are colored with a symbol in $\{\texttt{on}, \texttt{off}\}$ which is determined by the value of the counter on this cell. We call columns (resp.lines) colored with $\texttt{on}$ **computation-active** columns (resp. lines).

6.4.6.2 Adaptation of computing machines model to minimality property

In this section we present the way computing machines work in our construction. The model we use is adapted in order to have the minimality property, and is defined as follows:

Definition 6.10. *A* ***computing machine*** $\mathcal{M}$ *is some tuple* $= (\mathcal{Q}, \mathcal{A}, \delta, q_0, q_e, q_s, \#)$*, where* $\mathcal{Q}$ *is the state set,* $\mathcal{A}$ *the alphabet,* q_0 *the initial state, and* $\#$ *is the blank symbol, and*

$$\delta : \mathcal{A} \times \mathcal{Q} \to \mathcal{A} \times \mathcal{Q} \times \{\leftarrow, \to, \uparrow\}.$$

The other elements q_e, q_s *are states in* $\mathcal{Q}$*, such that for all* $q \in \{q_e, q_s\}$*, and for all* a *in* $\mathcal{A}$*,* $\delta(a, q) = (a, q, \uparrow)$*.*

The special states q_e, q_s in this definition have the following meaning:

- the error state q_e: a machine head enters this state when it detects an error, or when it collides with another machine head.

  This state is not forbidden in the subshift, but this is replaced by the sending of an error signal, and forbidding the coexistence of the error signal with a well initialized tape. The machine stops moving when it enters this state.

- shadow state q_s: because of multiple heads, we need to specify some state which does not act on the tape and does not interact with the other heads (acting thus as a blank symbol). The initial tape will have a head in initial state on the leftmost position and shadow states on the other ones.

Any Turing machine can be transformed into a such machine by adding some state q_s verifying the corresponding properties listed above.

Moreover, we add elements to the alphabet which interact trivially with the machine's states. This means that for any added letter a and any state q, $\delta(a, q) = (a, q, \uparrow)$, and then machines' states which interact trivially with the new alphabet, so that the cardinalities of the state set and the alphabet are equal to 2^{2^l}.

When the machine is well initialized, none of these states and letters will be reached. Hence the computations are the ones of the initial machine. As a consequence, one can consider that the machine we used has these properties.

In this section, we use a machine which does the following operations for all n

- write 1 on position p_n if $n = 2^k$ for some k and 0 if not.
- write $a_k^{(n)}$ on positions p_k, $k = 1...n$.

The sequence a is the Π_1-computable sequence defined at the beginning of the construction. The sequence $(a_k^{(n)})$ is a computable sequence such that for all k, $a_n = \inf_n a_k^{(n)}$. For all n, the position p_n is defined to be the number of the first a active column from left to right which is just on the right of an order n two dimensional cell on a face, amongst active columns.

6.4.6.3 Implementation of the machines

In this section, we describe the second sublayer of this layer.
Symbols:

The symbols are elements of the sets $\mathcal{A} \times \mathcal{Q}$, in $\mathcal{A}$, $\mathcal{Q}^2$, and a blank symbol.

Local rules:

- **Localization:** the non-blank symbols are superimposed on the bottom faces of the three-dimensional cells, according to direction $\mathbf{e}^3$. On these faces, they are superimposed on positions of computation-active columns and rows with p-counter value equal to $\overline{0}$ and border bit equal to 1. More precisely:
  - the possible symbols for computation active columns are elements of the sets $\mathcal{A}$, $\mathcal{A} \times \mathcal{Q}$ and elements of $\mathcal{A} \times \mathcal{Q}$ are on the intersection with computation-active rows.
  - other positions are superimposed with an element of $\mathcal{Q}^2$. See an illustration on Figure 6.24.
- along the rows and columns, the symbol is transmitted while not on intersections of computation-active columns and rows.

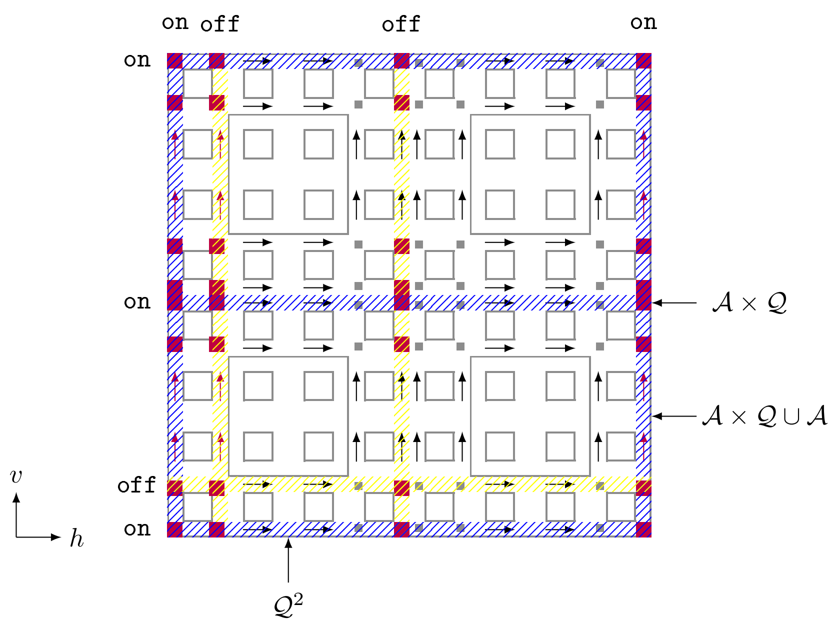


Figure 6.24: Localization of the machine symbols on the bottom faces of the cubes, according to the direction $\mathbf{e}^3$. Blue columns (resp. rows) symbolize computation-active columns (resp. rows).

- **Connection with the counter:** On active computation positions that are in the bottom line of this area, the symbols are equal to the corresponding subsymbol in $\mathcal{A} \times \mathcal{Q}$ on face 2 of the linear counter. On the leftmost (resp. rightmost) column of the area that are in a computation-active line, the symbol is equal to the corresponding symbol on face 7 (resp. 6) of the linear counter.
- **Computation positions rules:**

  Consider some computation position which is the intersection of a computation-active row and a computation-active column.

  For such a position, the **inputs** include:

  1. the symbols written on the south position (or on the corresponding position on face 2 when on the bottom line),
  2. the first symbol written on the west position (or the symbol on the corresponding position on face 7 when on the west border of the machine face),

3. and the second symbol on the east position (or the symbol on the corresponding position on face 6 when on the west border of the machine face).

The **outputs** include:

1. the symbols written on the north position (when not in the topmost row),
2. the second symbol of the west position (when not in the leftmost column),
3. and the first symbol on the east position (when not on the rightmost column).

Moreover, on the bottom line, the inputs from inside the area are always the shadow state q_s. See Figure 6.25 for an illustration.

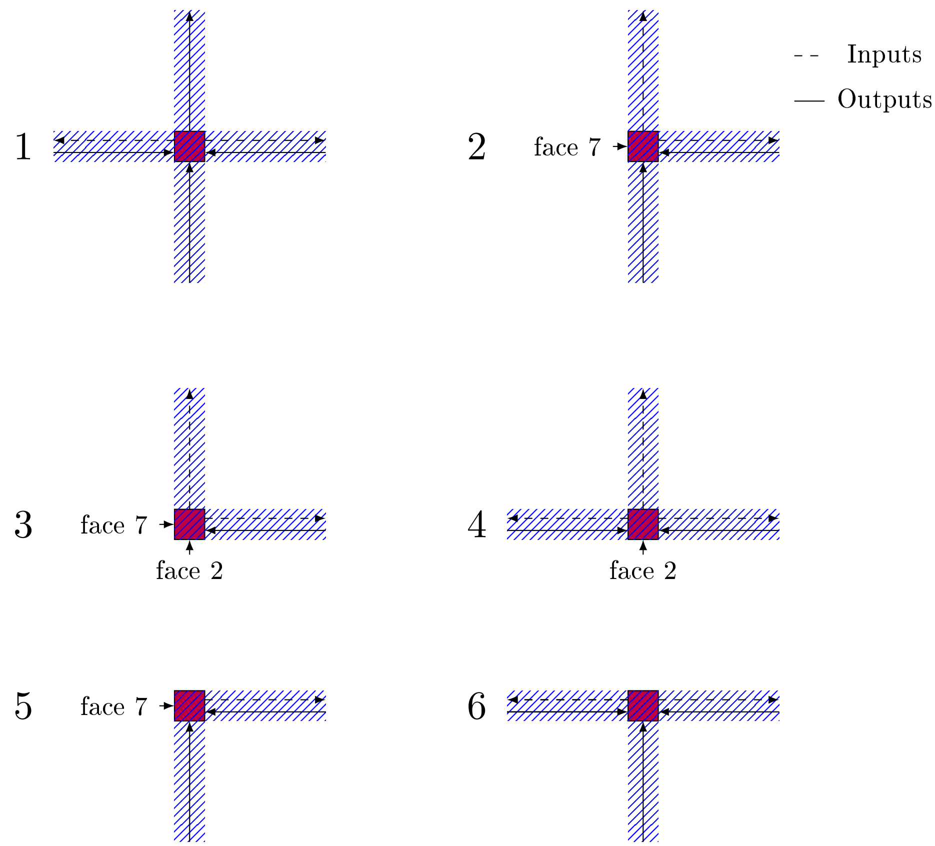


Figure 6.25: Schema of the inputs and outputs directions when inside the area (1) and on the border of the area (2,3,4,5,6).

On the first row, all the inputs are determined by the counter and by the above rule. Then, each computation-active row is determined by the adjacent one on the bottom and the value of the linear counter on faces 6 and 7, by the following rules. These rules determine, on each computation position, the outputs from the inputs:

1. **Collision between machine heads:** if there are at least two elements of $\mathcal{Q}\backslash\{q_s\}$ in the inputs, then the computation position is superimposed with (a, q_e). The output on the top

(when this exists) is (a, q_e), where a is the letter input below. The outputs on the sides are q_s. When there is a unique symbol in $\mathcal{Q} \backslash \{q_s\}$ in the inputs, this symbol is called the machine head state (the symbol q_s is not considered as representing a machine head).

2. **Standard rule:**

   (a) when the head input comes from a side, then the functional position is superimposed with (a, q). The above output is the ordered pair (a, q), where a is the letter input under, and q the head input. The other outputs are q_s. See Figure 6.26 for an illustration of this rule.

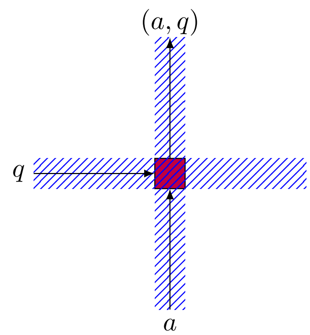


Figure 6.26: Illustration of the standard rules (1).

   (b) when the head input comes from under, the above output is:
   - $\delta_1(a, q)$ when $\delta_3(a, q)$ is in $\{\rightarrow, \leftarrow\}$
   - and $(\delta_1(a, q), \delta_2(a, q))$ when $\delta_3(a, q) = \uparrow$.

   The head output is in the direction of $\delta_3(a, q)$ (when this output direction exists) and equal to $\delta_2(a, q)$ when it is in $\{\rightarrow, \leftarrow\}$. The other outputs are q_s. See Figure 6.27 for an illustration.

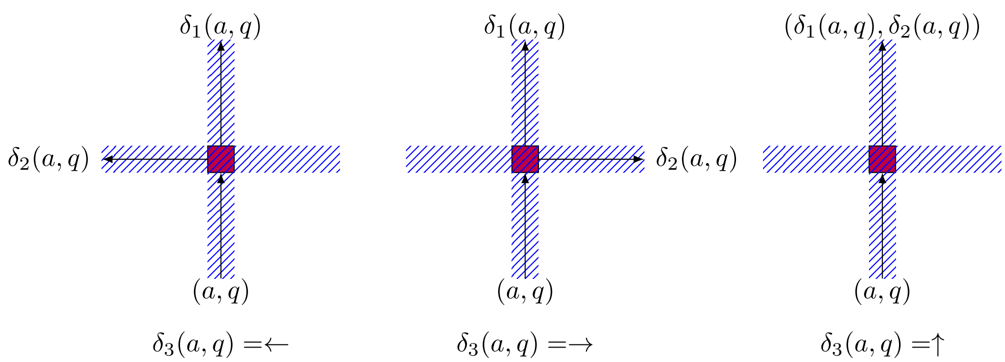


Figure 6.27: Illustration of the standard rules (2).

3. **Collision with border:** When the output direction does not exist, the output is (a, q_e) on the top, and the outputs on the side is q_s. The computation position is superimposed with (a, q).

4. **No machine head:** when all the inputs in $\mathcal{Q}$ are q_s, and the above output is in $\mathcal{A}$ and equal to its input a.

Global behavior:

On the bottom faces according to $\mathbf{e}^3$ of order qp three-dimensional cells, we implemented some computations using our modified Turing machine model. This model allows multiple machine heads on the initial tape and entering in each row. When there is a unique machine head on the leftmost position of the bottom line and only blank letters on the initial tape, and all the lines and columns are computation-active, then the computations are as intended. This means that the machine writes successively for all n, pairs of bits on the p_kth column of its tape (in order to impose the value of the frequency bits), for $k = 1...n$. The first bit imposes a constraint on the frequency bit and is equal to $a_k^{(n)}$, and the second bit imposes a constraint on the grouping bit, and is equal to 1 if k is a power of two, and 0 if not. Each time it erases the previous bit and writes over it. The machine enters in the error state q_e when it detects an error.

When this is not the case, the computations are determined by the rules giving the outputs on computation positions from the inputs. When there is a collision of a machine head with the border, it enters state q_e. When heads collide, they fusion into a unique head in state q_e. In Section 6.4.7, we describe signal errors that helps us to take into account the computations only when the initial tape is empty, all the lines and columns are computation-active, and there is no machine head entering on the sides of the area.

Although computation does not end at a given level, the only important thing is that any step is reached by a machine in a sufficiently great level.

6.4.7 Error signals

In order to simulate any behavior that happens in infinite areas in finite ones, we need error signals. This means that when the machine detects an error (enters a halting state), it sends a signal to the initialization line to verify it was well initialized: that the tape was empty, that no machine head enters on the sides, and that the machine was initially in the leftmost position of the line, and in initialization state. Moreover, for the reason that we need to compute precisely the number of possible initial tape contents, we allow initialization of multiple heads. The first error signal will detect the first position from left to right in the top row of the area where there is a machine head in error state or the active column is `off`. This position only will trigger an error signal (described in Section 6.4.7.3, according to the direction specified just above when it in the top line of the area (the word of arrows specifying the direction is a part of the counter). The empty tape signal detects if the initial tape was empty, and that there was a unique machine head on the leftmost position in initialization state q_0. The empty tape and first error signals are described in Section 6.4.7.1. The empty sides signal, described in Section 6.4.7.2 detects if there is no machine head entering on the sides, and that the `on`/`off` signals on the sides are all equal to `on`. The error signal is taken into account (meaning forbidden) when the empty tape, and empty sides signals are detecting an error.

6.4.7.1 Empty tape, first error signals

Symbols:

The first sublayer has the following symbols:
■,■, symbols in $\{$■,■$\}^2$, and a blank symbol □.

Local rules:

- **Localization:** non blank symbols are superimposed on the top line and bottom line of the border of the machine face as a two-dimensional cell.

- **First error signal:** this signal detects the first error on the top of the functional area, from the left to the right, where an error means a symbol `off` or q_e. The rules are:

  - the topmost leftmost position of the top line of the cell is marked with .
  - the symbol propagates the the left, and propagates to the right while the position under is not in error state q_e and the symbol in $\{\texttt{on}, \texttt{off}\}$ is `on`.
  - when on position in the top row with an error, the position on the top right is colored .
  - the symbol propagates to the right, and propagates to the left while the positions under is not in error state.

  Empty tape signal: this signal detects if the initial tape of the machine is empty. This means that it is filled with the symbol $(\#, q_s)$ except on the leftmost position where it has to be $(\#, q_0)$. The signal detects the first symbol which is different from $(\#, q_s)$ or $(\#, q_0)$ when on the left, from left to right (first color), and from left to right (second color). Concerning the first color:

  - on the bottom row, the leftmost position is colored with .
  - The symbol propagates to the right unless when on a position under a symbol different from:
    - $(\#, q_0)$ when on the leftmost functional position,
    - $(\#, q_s)$ on another functional position.
  - When on these positions, the symbol on the right is .
  - the symbol propagates to the right.

  for the second one:

  - on the bottom row, the rightmost position is colored with .
  - The symbol propagates to the left except when on a position under a symbol different from:
    - $(\#, q_0)$ when on the leftmost functional position,
    - $(\#, q_s)$ on another computation position.
  - When on these positions, the symbol on the right is .
  - the symbol propagates to the left.

Global behavior:

The top row is separated in two parts: before and after (from left to right) the first error. The left part is colored , and the right part . The bottom row is colored with a ordered pair of colors. The first one separates the row in two parts. The limit between the two parts is the first occurrence *from left to right* of a symbol different from $(\#, q_s)$ or $(\#, q_0)$ when on the leftmost computation position above. The second color of the ordered pair separates two similar parts *from right to left*.

See Figure 6.28 for an illustration.

6.4.7.2 Empty sides signals

This second sublayer has the same symbols as the first sublayer. The principle of the local rules is similar: the leftmost (resp. rightmost) column is split into two parts, the top one colored and the bottom one colored . The limit is the first position from top to bottom where the corresponding symbol across the limit with face 7 (resp. face 6) is $(q_s, \texttt{on})$ (resp. q_s). Moreover, the bottom row of the machine face is colored with an ordered pair of colors. This ordered pair is constant over the row. The first one of the colors is equal to the color at the bottom of the leftmost column. The second one is the color at the bottom of the rightmost one.

See an illustration on Figure 6.28.

6.4.7.3 Error signals

Symbols:

 (error signal), .

Local rules:

- **Localization:** the non-blank symbols are superimposed on the right, left and top sides of the border of machine faces in three-dimensional cells.
- **Propagation:** each of the two symbols propagates when inside one of these two areas:
  - the union of the left side of the face and positions colored in the top side of the face.
  - the union of the right side of the face and positions colored in the top side of the face.
- **Induction:**
  - on a position of the top side of the face which is colored and the position on the right is , if the symbol above in the information transfers layer is $\rightarrow$ (resp. $\leftarrow$), then there is an error signal on this position and none on the right (resp. there is no error signal on this position and there is one on the right).
  - on the rightmost topmost position of the face, if the first machine signal is , then there is no error signal.
- **Forbidding wrong configurations:**

  there can not be four symbols and an error signal on the same position.

Global behavior:

When there is a machine head in error state in the top row, the first one (from left to right) sends an error signal to the bottom row (see Figure 6.28) in the direction indicated by the arrow on the corresponding position on face 3. This signal is forbidden if the machine is well initialized. This means that the working tape of the machine is empty in the bottom row. Moreover, there is a unique machine in state q_0 in the leftmost position of the bottom row, all the lines and columns are $\texttt{on}$ and there is no machine entering on the sides. This means that the error signal is taken into account only when the computations have the intended behavior.

Because for any n and any configuration, there exists some three-dimensional cell in which the machine is well initialized (because of the presence of counters). For any k, there exists some n such that the machine has enough time to check the kth frequency bit and grouping bit. This means that in any configuration of the subshift, the kth frequency bit is equal to a_k, and the 2^kth grouping bit is 1, and the other ones are 0.

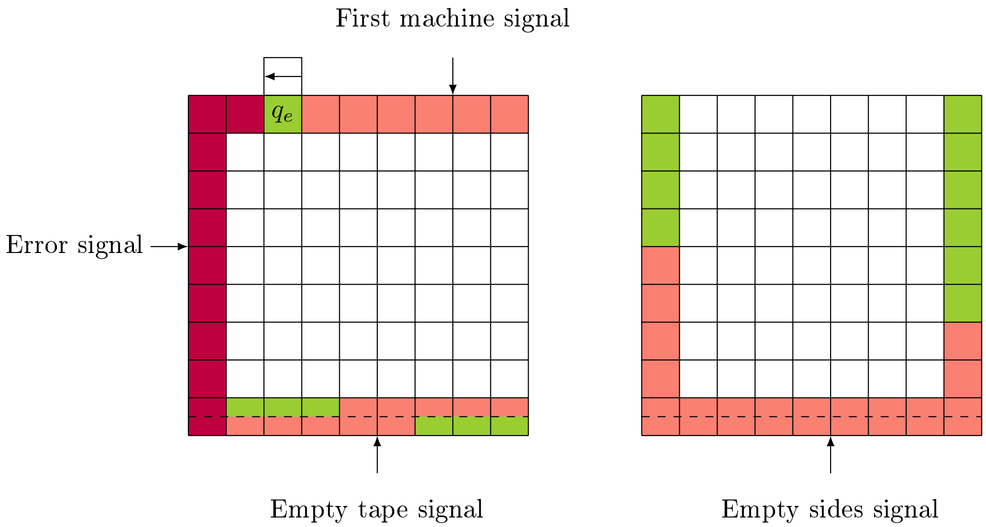


Figure 6.28: Illustration of the propagation of an error signal, where are represented the empty tape, first machine and empty sides signals.

6.4.8 Hierarchical counter layer

This layer has two sublayers.

6.4.8.1 Grouping border bits

This sublayer supports bits on the top faces of the three-dimensional cells, according to $\mathbf{e}^1$, on positions that are not in the border of a two-dimensional cell with grouping bit bit 1. The symbol are $0, 1$ and propagate to neighbors while these neighbors have not grouping bit equal to 1. When near the border of a cell with grouping bit equal to 1, the bit is 1 if it is the border of the face of the three-dimensional cell. Else, this bit is 0.

This means that the only functional positions which have bit 1 are the functional position on a face of an order $2^k p$ three-dimensional cell which are not in an order $2^j p$ two-dimensional cell with $j < k$.

We use a similar mechanism as the one presented in Section 6.4.2.1 so that **the order n two dimensional cells have access to the grouping bit and the frequency bit of the two-dimensional order $n+1$ cell just above in the hierarchy, as well as its p-addresses** (for the addresses there is not border rule or ϵ symbol).

6.4.8.2 Hierarchy bits

Symbols:

, , elements of $\{ , \}^2$ and a blank symbol.

Local rules:

- **Localization:** the non-blank symbols of this layer are localized on the petal positions of the copy of the Robinson subshift which is parallel to $\mathbf{e}^2$ and $\mathbf{e}^3$.

- the ordered pairs are superimposed on and only on transformation positions.

- the symbol is transmitted through arrows and intersection of order n and $n+1$ petals such that the order n one has counter 0 or the value of the p-counter is not $\overline{p-1}$. The other intersection positions are **transformation positions**.

- **Transformation:** on a transformation position, we have the following rules:

  – if the grouping bit of the above cell in the hierarchy is 1, then we have the following:

    * in the following cases, the second color is :

      1. the Robinson symbol is , or , and the orientation symbol is ,
      2. the Robinson symbol is , or , and the orientation symbol is ,
      3. the Robinson symbol is , or , and the orientation symbol is .
      4. the Robinson symbol is , or , and the orientation symbol is .

    * in the following cases, the second color is equal to the first one:

      1. the Robinson symbol is , or , and the orientation symbol is ,
      2. the Robinson symbol is , or , and the orientation symbol is ,
      3. the Robinson symbol is , or , and the orientation symbol is .
      4. the Robinson symbol is , or , and the orientation symbol is .

    * in the other cases, the second color is .

  See an illustration of Figure 6.29.

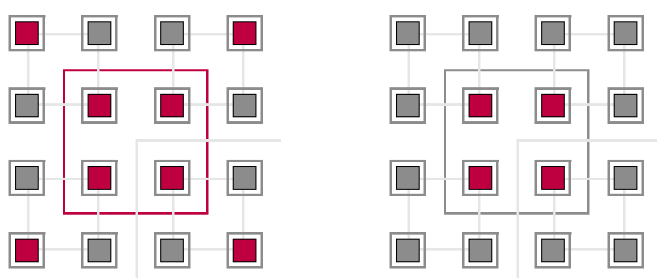


Figure 6.29: Illustration of the transformation rules of the hierarchy bits when the grouping bit is 1.

  – when:

    * grouping bit of the cell above in the hierarchy is 0,
    * the value of the p-counter is $\overline{0}$,
    * and its frequency bit is 0,

  then the transformation rule relies on the addresses. If the ordered pair of addresses is in

$$\{(1^p, 0^p), (0^p, 1^p), (0^p, 0^p), (1^p, 1^p)\},$$

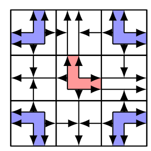
  then the color is transmitted. In the other cases, it becomes

– when:
  * the grouping bit of the cell above in the hierarchy is 0,
  * the value of the p-counter is $\overline{0}$ and the frequency bit is 1,
  * or the value of the p-counter is not $\overline{0}$,

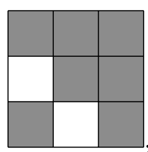
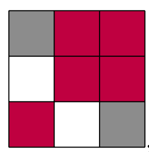
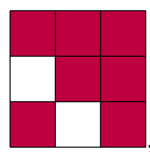
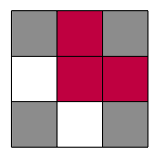
  then the second color is equal to the first one.

– **Coherence rules:** since the hierarchy bits are superimposed on the copy of the Robinson subshift parallel to $\mathbf{e}^2$ and $\mathbf{e}^3$, the degenerated behaviors of this layer correspond to the ones of the Robinson subshift. In this list we consider patterns intersecting two-dimensional cells having order greater or equal to p.

  1. when near a **corner**, on a pattern

     the pattern in this layer has to be amongst the following ones:

     , , , 

     respectively when
     (a) the central corner is colored gray,
     (b) this corner is purple and the grouping bit is 1.
     (c) this corner is purple, the grouping bit is 0 and else the p-counter has not value $\overline{0}$, or this value is $\overline{0}$ and the frequency bit is 1, or the addresses are in

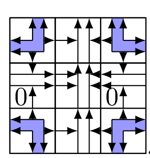
     $$\{(0^p, 0^p), (0^p, 1^p), (1^p, 0^p), (1^p, 1^p)\}.$$

     (d) this corner is purple, the grouping bit is 0, the p-counter has value $\overline{0}$, the frequency bit is 0, and the addresses are not in

     $$\{(0^p, 0^p), (0^p, 1^p), (1^p, 0^p), (1^p, 1^p)\}.$$

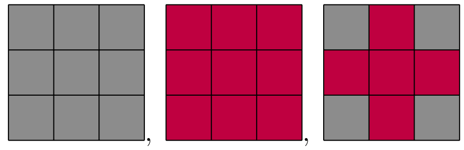
     There are similar rules for the other corners. This allows degenerated behaviors which do not happen around finite cells to be avoided. The other rules in this list have the same function.

  2. near a the middle or the quarter of an edge of a cell or near a corner of a petal with $0,1$-counter value equal to 0, the symbols on the blue corners are gray if the border of the cell is gray. If this is purple, the grouping bit is 0, the p-counter has value $\overline{0}$ and the frequency bit is 0, and the addresses are not in $\{(0^p, 0^p), (0^p, 1^p), (1^p, 0^p), (1^p, 1^p)\}$, the blue corners are colored gray. In the other cases, they are colored purple. For instance, on the pattern

     ,

the pattern has to be amongst the following ones:



Global behavior:

Over the copy of the Robinson subshift which is parallel to $\mathbf{e}^2$ and $\mathbf{e}^3$, we make a signal propagate and transform on intersections of support petals and the transmission petal just above in the hierarchy. With this signal, we color the blue corners with a hierarchy bit in {,}. This process relies on the frequency bits in a similar way as in the proof of Theorem 3.19, except that each time the signal enters inside an order $2^k p$ two-dimensional cell, not matter the color of the cell, the support petals just under in the hierarchy are colored . This leads to the preservation of the minimality property. After synchronization, Lemma 6.11 count the cardinality of these sets.

6.4.8.3 Random bits

Symbols:

Elements of $\{0,1\}$ (these are the random bits, which generate entropy dimension) and a blank symbol.

Local rules:

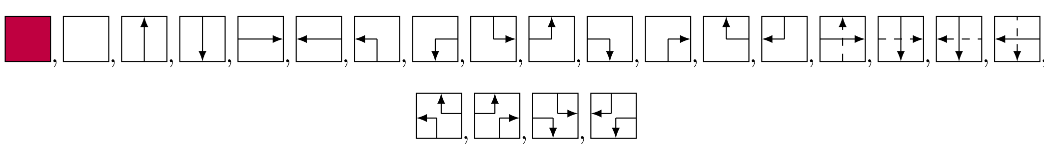
- **Localization:** the non blank symbols are superimposed on and only on positions with north west blue corner having a hierarchy bit equal to .
- **Transformation:** for any position $\mathbf{u} \in \mathbb{Z}^3$ which is not on the top (according to $\mathbf{e}^1$) face of a three-dimensional cell, the random bit on position $\mathbf{u}$ is equal to the random bit on position $\mathbf{u}+\mathbf{e}^1$. The transformation rule on these faces will be stated in Section 6.4.8.5.

The next sections describe the set of random bits as a value (similarly to the linear counter) incremented by 1 going through the top face of a three-dimensional cell according to direction $\mathbf{e}^1$.

6.4.8.4 Convolutions

In this sublayer we generate a path on the top faces of the three-dimensional cells according to direction $\mathbf{e}^1$.

Symbols:



Local rules:

- **Localization:** the non-blank symbols are superimposed only on the top faces of the three-dimensional cells according to direction $\mathbf{e}^1$.

- **Transmission:** the arrows and blank symbols propagate through five or four arrows symbols of the Robinson copy parallel to $\mathbf{e}^2$ and $\mathbf{e}^3$.

- **Triggering positions:**

  The position on the top or bottom line of the border of the face, with six arrows symbol in the Robinson copy parallel to this face is called **triggering position**. These positions are superimposed with , and this symbol can be only on this position. The presence of this symbol implies a right arrow symbol on the position on the right, a left arrow on the left and a blank symbol above if on the top line. When on the bottom line, the direction of the arrows is reversed, and the blank symbol is forced upwards.

- **Correspondence for other types of positions:**

  The following tables give a correspondence between other types of positions and the symbol superimposed on them in this layer. Table 6.1 is related to positions on the border of a face. In this table, induction refers to the fact that the position type imposes the presence of a symbol on a position nearby. The symbol and the position are specified in the table. Table 6.2 refers to positions inside a face.

- **Determination of symbols on the border:**

  A cross symbol can not be present on the border.

Type of position	Symbol	Inducted symbols	Inducted positions
Corners of the area			
On the top line, superimposed with		,	right, resp. left position
Top line,		,	right, resp. left position
Leftmost column,		,	bottom, resp. top position
Leftmost column,		,	bottom, resp. top position
Rightmost column,		,	bottom, resp. top position
Rightmost column,		,	bottom, resp. top position
Bottommost line,		,	right, resp. left position
Bottommost line,		,	right, resp. left position

Table 6.1: Correspondence table for positions on the **border** of a face.

Global behavior:

The global behavior is the drawing of a curve starting and ending at some position at the center of the top or bottom row of the border (depending on the orientation of the three-dimensional cell) of the top face according to $\mathbf{e}^1$ of three-dimensional cells. This position is specified by purple color, and the path is going through every position having a random bit on it. This serves as a circuit for the incrementation signal of the hierarchical counter. See Figure 6.35.

6.4.8.5 Incrementation signal

Symbols:

This sublayer has symbols in $\{$,$\}$ or $\{$,$\}^2$ and a blank symbol.

Type of position	Symbol	Type of position	Symbol
Superimposed with			
	or		
	or		
	or		
	or		

Table 6.2: Correspondence table for positions **inside** a face.

Local rules:

- **Localization :** the non blank symbols are superimposed only on non blank convolution symbols when on the top face according to $\mathbf{e}^1$ of a three-dimensional cell.

- when outside these faces, the non-blank symbols are superimposed on the positions that don't have a petal symbol, and the symbol on a position $\mathbf{u} \in \mathbb{Z}^3$ is transmitted to next positions $\mathbf{u} \pm \mathbf{e}^2$ and $\mathbf{u} \pm \mathbf{e}^3$ when these positions are not in the border of a two-dimensional cell with grouping bit equal to 1. These positions have a simple symbol.

- the two bits symbols can be superimposed only on positions with a cross symbol in the convolution sub-sublayer, and on top faces of three-dimensional cells according to $\mathbf{e}^1$.

- **Initialization:** The position with purple symbol in the convolution sublayer is superimposed with .

- **Transmission:** the color is transmitted to the next position in the direction of the arrow in the convolution sublayer if:

  - this next position has random bit equal to 1 or the grouping border bit is 0 (we use this rule so that the counter in a $2^k p$ order two-dimensional cell does not interact with the random bits that are in a $2^j p$ order cell, with $j < k$),
  - or the next position has the purple symbol in the convolution sublayer.

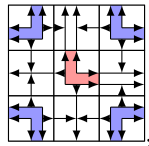
  The symbol is changed only in these cases. In other cases, the next position is marked with .

- the symbol is transmitted through the arrow unless the next position has the purple symbol.

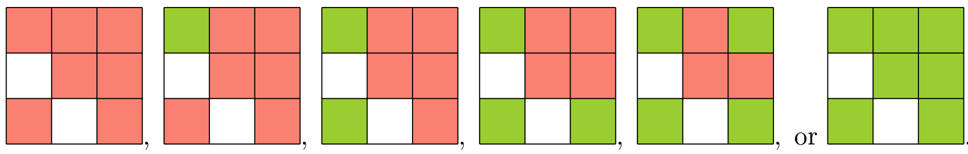
- **Freezing signal:** on the position $\mathbf{u}$ with purple symbol in the convolution sublayer, if the color on the next position with arrow pointing on this position is , then the symbols on positions $\mathbf{u} - \mathbf{e}^1$ and $\mathbf{u} + \mathbf{e}^1$ are different. Else, they are equal. When a position on the face have symbol , then the symbol on positions $\mathbf{u} - \mathbf{e}^1$ and $\mathbf{u} + \mathbf{e}^1$ is .

- **Incrementation:** on position $\mathbf{u}$ of a top face according to $\mathbf{e}^1$, if the grouping bit is 1 and the grouping border bit is 1, the symbol is . The symbol on position $\mathbf{u} + \mathbf{e}^1$ is , the random bit on position $\mathbf{u} + \mathbf{e}^1$ is 0. Else, it is equal to the random bit on position $\mathbf{u}$.

- **Coherence rule:** we use coherence rules so that on degenerated behaviors of the structure layer, the sense of propagation of the incrementation signal is respected. For instance, on a pattern inside the face being

,

the possible colorings are:



Global behavior:

When going through an order $n \neq 2^k p$ three-dimensional cell for any k, the random bits are not changed. When going through the top face according to $\mathbf{e}^1$ of an order $2^k p$ order three-dimensional cell, the random bits that are in the corresponding two-dimensional cell and are not in a $2^j p$ cell with $j < k$ are grouped (using the grouping bits and the grouping border bits). The sequence of these bits is incremented by 1 - in a similar way as is incremented the linear counter, using the convolutions - when the freezing signal is . It is not incremented when it is . This last case happens only once when this set of bits is in maximal position. Because of this, and the formula proved in the first point of Lemma 6.11, the period of this counter is $2^{2^{l_k}} + 1$, where

$$\boxed{l_k = 4[(k+1)(p-1)+1] + 2(2^k - 1 - c_k) + 4pc_k}$$

The functions $k \mapsto 2^k - 1 - c_k$ and $k \mapsto c_k$ are non-decreasing, and the function $k \mapsto (k+1)(p-1)$ is strictly increasing. Hence all the numbers l_k are different. As a consequence, all these numbers are different Fermat numbers.

6.4.9 Synchronization layer

We use this layer to synchronize the hierarchical counters in the orthogonal directions of their incrementation direction $\mathbf{e}^1$ (we recall that from the way we coded the linear counters, they are already synchronized in the orthogonal directions of their incrementation direction $\mathbf{e}^2$) for three-dimensional cells having the same order in case of the linear counter. We synchronize only $2^k p$ cells for the same $k \geq 0$ (this is the purpose of grouping bits).

This layer has two sub-layers.

6.4.9.1 Synchronization areas

The aim of this first sublayer is to localize places where synchronization of hierarchical counters occurs.
Symbols:

The symbols are $(\boxslash, \square)^2$.

Local rules:

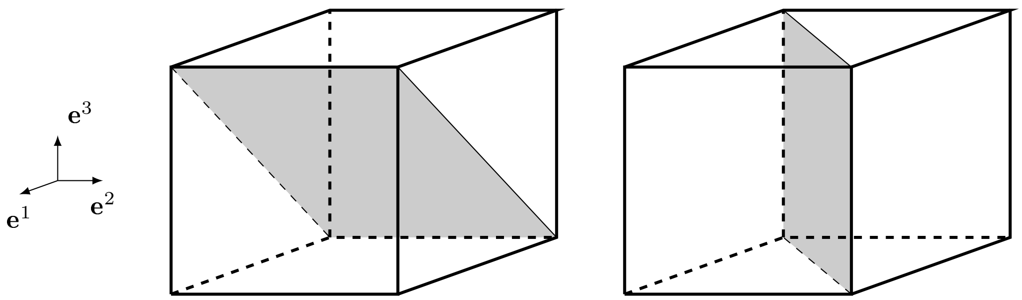


Figure 6.30: Illustration of the synchronization areas rules. The plane generated by the first symbol of the ordered pair is on the left, and the other one on the right.

- the first symbol of the ordered pair is transmitted in the directions $\pm\mathbf{e}^2$, and the second one in the directions $\pm\mathbf{e}^3$.
- on a position $\mathbf{u} \in \mathbb{Z}^3$ which is not superimposed with a corner in the copy of the Robinson subshift parallel to $\mathbf{e}^3$ and $\mathbf{e}^1$, the first symbol of the ordered pair is transmitted to positions $\mathbf{u}+\mathbf{e}^3+\mathbf{e}^1$ and $\mathbf{u}-\mathbf{e}^3-\mathbf{e}^1$.
- on a position $\mathbf{u} \in \mathbb{Z}^3$ which is not superimposed with a corner in the copy of the Robinson subshift parallel to $\mathbf{e}^1$ and $\mathbf{e}^2$, the first symbol of the ordered pair is transmitted to positions $\mathbf{u}+\mathbf{e}^1+\mathbf{e}^2$ and $\mathbf{u}-\mathbf{e}^1-\mathbf{e}^2$.
- the symbol $\boxslash$ as first symbol of the ordered pair can not be superimposed on a position in the border of of a two-dimensional cell in the Robinson copy parallel to $\mathbf{e}^3$ and $\mathbf{e}^1$ except a corner.
- the symbol $\boxslash$ as second symbol of the ordered pair can not be superimposed on a position in the border of of a two-dimensional cell in the Robinson copy parallel to $\mathbf{e}^1$ and $\mathbf{e}^2$ except a corner.
- on a position $\mathbf{u}$ with a north east or south west corner in the Robinson copy parallel to $\mathbf{e}^3$ and $\mathbf{e}^1$, the first symbol of the ordered pair is $\boxslash$. According to the orientation of the corner, it forces the presence of another symbol $\boxslash$ on another position as follows:

  1. if a north east corner, the symbol is forced on position $\mathbf{u}-\mathbf{e}^3-\mathbf{e}^1$
  2. if a south west corner, it is forced on $\mathbf{u}+\mathbf{e}^3+\mathbf{e}^1$.

  Moreover, when the grouping bit is 0, it forces the symbol $\boxslash$ on positions:

  1. if a north east corner, the symbol is forced on position $\mathbf{u}+\mathbf{e}^3+\mathbf{e}^1$
  2. if a south west corner, it is forced on $\mathbf{u}-\mathbf{e}^3-\mathbf{e}^1$.
- on a position $\mathbf{u}$ with a north east or south west corner in the Robinson copy parallel to $\mathbf{e}^3$ and $\mathbf{e}^1$, the second symbol of the ordered pair is $\boxslash$. According to the orientation of the corner, it forces the presence of another symbol $\boxslash$ on another position as follows:

  1. if a north east corner, the symbol is forced on position $\mathbf{u}-\mathbf{e}^1-\mathbf{e}^2$
  2. if a south west corner, it is forced on $\mathbf{u}+\mathbf{e}^1+\mathbf{e}^2$.

  Moreover, when the grouping bit is 0, it forces the symbol $\boxslash$ on positions:

  1. if a north east corner, the symbol is forced on position $\mathbf{u}+\mathbf{e}^1+\mathbf{e}^2$

2. if a south west corner, it is forced on $\mathbf{u} - \mathbf{e}^1 - \mathbf{e}^2$.

Global behavior:

The global behavior induced by these rules is the drawing, for any three-dimensional cell having order $2^k p$ for some $k \geq 0$, of two planes crossing it. One of them links the north east edge parallel to $\mathbf{e}^2$ to the south west one. The other one links the north east edge parallel to $\mathbf{e}^3$ to the south west one, as on Figure 6.30. The first plane is drawn by the first symbol and generated by the vectors $\mathbf{e}^3 + \mathbf{e}^1$ and $\mathbf{e}^2$. The other one is drawn by the second symbol and is generated by the vectors $\mathbf{e}^1 + \mathbf{e}^2$ and $\mathbf{e}^3$.

For all k, these planes cross order j cells for all $j < 2^k p$ that are inside the qp order three-dimensional cell. However, an order $j \neq 2^k p$ cell for any k, one of these planes can cross this cell if and only if it is on the diagonal plane of an order $2^k p$ cell.

6.4.9.2 Synchronization of the hierarchical counters

This sub-sublayer consists of two copies of the random bit sublayer, the hierarchy bit sublayer and the incrementation signal sublayer, except that the first set of copies is parallel to $\mathbf{e}^1$ and $\mathbf{e}^2$ and the second one to $\mathbf{e}^1$ and $\mathbf{e}^3$ (instead of $\mathbf{e}^2$ and $\mathbf{e}^3$ for the hierarchical counter). Moreover, in these copies, there is no incrementation signal. This means that the rules on the faces of three-dimensional cells are the same as those on the outside. Furthermore, when the first symbol of the synchronization areas sublayer is $\boxslash$, then the random bit in the first copy is equal to the random bit of the hierarchical counter. When the second symbol of the synchronization areas sublayer is $\boxslash$, then the random bit in the second copy is equal to the random bit of the hierarchical counter.

The *global behavior* induced is that the values of the hierarchical counter in two adjacent, in else direction $\mathbf{e}^2$ or $\mathbf{e}^3$, order $2^k p$ three-dimensional cells, are equal.

As a consequence, one can state the following lemma:

Lemma 6.11. *For all $q \neq 2^k$ for any k, the number r_q of possible colorings by hierarchy bits and random bits on a $2qp+2$ order supertile (centered on an order qp cell) verifies the following inequalities:*

$$\boxed{\left[\alpha(p).q^{\lambda(p)}\right] 4^{c_q^0}.16^{pc_q^1} \geq \log_2(r_q) \geq 4^{c_q^0}.16^{pc_q^1}}$$

where $c_{q,1}$ is the number of $i \leq q-1$ such that $a_i = 1$ and $c_{q,0}$ is the number of $i \leq q-1$ such that $a_i = 0$, and $\alpha(p), \lambda(p) > 0$ depend only on p.

Proof. • **A formula on the number of proper blue corner positions in a cell having hierarchy bit $\blacksquare$:**

For all $k \geq 0$, in an order $2^k p$ cell, the number of blue corners having hierarchy bit $\blacksquare$ that are not in an order $2^j p$ cell, $j < k$, is

$$d_k = 4.4^k.16^{k(p-1)}.4^{2^k-1-c_k-k}.4.16^{p-1}.16^{pc_k},$$

where c_k is the number of $i \leq 2^k - 1$ such that the ith frequency bit a_i is 1.

Indeed:

1. the process that rules the coloring with hierarchy bits of this cell starts on 4 order $2^k p - 1$ cells, hence the factor 4.16^{p-1}.
2. The factor $4^k.16^{k(p-1)}$ comes from the transitions occurring when the grouping bit is 1 (there are k such transitions).

3. The factor $4^{2^k-1-c_k-k}$ comes from the transitions occurring when the grouping bit is 0 and the frequency bit is 0.
4. The factor 4.16^{pc_k} comes from the transitions when the grouping bit is 0 and the frequency bit is 1.
5. The factor 4 comes from the fact that there are four blue corners for each order 0 cell.

- **On the hierarchy bits of a supertile:**

  An order $2qp+2$ supertile is centered on an order qp cell. There are two possibilities for the hierarchy bits, as follows.

  1. **The border of this cell is colored ■.** This means that all the blue corners in the supertiles colored with ■, except the ones that are in an order $2^j p$ cell with $2^j \leq q$. This is equivalent to $j \leq \lfloor \log_2(q) \rfloor$. Since the jth hierarchical counter for all $j \leq \lfloor \log_2(q) \rfloor$ in this supertile are synchronized, the number of possible colorings of this supertile by random bits is given by the product for $j \leq \lfloor \log_2(q) \rfloor$ of the numbers of possible colorings by random bits of the set of blue corners that are in an order $2^j p$ cell and not in an order $2^i p$ cell with $i < j$. These numbers are 2^{d_j}, so the total number of random bits displays is $2^{\lambda_1(q)}$, where

  $$\lambda_1(q) = \sum_{j=0}^{\lfloor \log_2(q) \rfloor} d_j.$$

  2. **The border is colored ■.** In this case, we also have to count the number of hierarchy bits that are in the supertile but not in an order $2^j p$ cell with $j \leq \lfloor \log_2(q) \rfloor$. This case is similar to the first point of this proof, and the number of random bits displays in this case is $2^{\lambda_2(q)}$, where:

  $$\lambda_2(q) = \sum_{j=0}^{\lfloor \log_2(q) \rfloor} d_j + 4.4^{\lfloor \log_2(q) \rfloor + 1}.16^{\lfloor \log_2(q) \rfloor (p-1)}.4^{c_q^0 - \lfloor \log_2(q) \rfloor}.16^{pc_q^1}.$$

  As a consequence, the total number of possibilities for the hierarchy bits and random bits is

  $$2^{\lambda_1(q)} + 2^{\lambda_2(q)} \leq 2.2^{\lambda_2(q)},$$

  since $\lambda_1(q) \leq \lambda_2(q)$.

- **Inequalities:**

  The lower bound is clear. For the upper bound, we have, following the last point:

  $$\log_2(r_q) \leq \left(4.4^{\lfloor \log_2(q) \rfloor + 1}.16^{\lfloor \log_2(q) \rfloor (p-1)}.4^{c_q^0 - \lfloor \log_2(q) \rfloor}.16^{pc_q^1} + \sum_{j=0}^{\lfloor \log_2(q) \rfloor} d_j + \right) + 1$$

  Hence from the fact that the functions $k \mapsto c_k$ and $k \mapsto 2^k - 1 - c_k$ are non-decreasing and the inequality $\log_2(q) \geq \lfloor \log_2(q) \rfloor$:

  $$\begin{array}{rcl} \log_2(r_q) & \leq & 2(\log_2(q)+1).4.4^{\log_2(q)+1}.16^{\log_2(q)(p-1)}.4^{c_q^0 - \log_2(q)+1}.16^{pc_q^1} \\ & = & 2(\log_2(q)+1).4^2.16^{\log_2(q)(p-1)}.4^{c_q^0}.16^{pc_q^1} \end{array}.$$

  Since $2(\log_2(q)+1).4^2.16^{\log_2(q)(p-1)}$ can be bounded by $\alpha(p) q^{\lambda(p)}$ for some functions α and λ, this provides the upper bound.

$\square$

6.5 Properties of the subshifts X_z:

6.5.1 Pattern completion

In this section, we prove the following proposition, which will serve to prove that X_z is minimal. We assume the reader has some familiarity with the properties of the Robinson subshift.

Proposition 6.12. *Let P some n-block in the language of X_z. Then p can be completed into an admissible pattern over a three-dimensional cell.*

Let P some n block in the language of X_z which appears in some configuration x. Let us prove that it can be completed into a three-dimensional supertile. We follow some order in the layers for the completion. First we complete the pattern in the structure layer, then in the functional areas layer, the linear counter and machine layers and then the hierarchy bits and the hierarchical counter. After this we prove how this supertile can be completed into a three-dimensional cell.

6.5.1.1 Completion of the structure

When the pattern P is a sub-pattern of an infinite supertile of x, this is clear that the projection of P into the structure layer can be completed into a finite supertile in the configuration x.

When the support of P crosses the separating area between the supports of the infinite supertiles, as in the proof of Proposition 3.28, we can still complete the projection of P over the structure layer into an order k three-dimensional supertile, with k great enough.

6.5.1.2 Completion of the functional areas

The completion in the functional areas layer is more subtle and depends on how the support of the pattern crosses the area between the supports of infinite supertiles. We will choose k great enough in each of the cases. We tell how to complete the border of the greatest three-dimensional cell included in the supertile, since the smaller cells can be completed according to the configuration x. Here is a list of all the possible cases:

- When the pattern intersects the corner of an infinite three-dimensional cell, we complete the projection of P on the structure layer on the inside of each of the faces into the inside of a two-dimensional cell (without the border). The size of the two-dimensional cell is chosen according to the value of the p-counter and the grouping bit of the three-dimensional cell corner. The coherence rules allow the coloring of the petals in this two-dimensional cell to be determined. Indeed, since this is the corner of a three-dimensional cell, the border rules imply that the borders of the faces are colored with in the functional areas sublayer, and in the active areas sublayer. The coherence rules allow the colors of the blue corners around the corner of the three-dimensional infinite cell to be determined. The transformation rules imply that there is a unique possibility for the coloring of the border of the order 0 two-dimensional cells around, then the order 1 two-dimensional cells, etc. This thus determines the color of the other petals in the inside of the two dimensional cells. See an illustration on Figure 6.31 for the functional areas sublayer.

  The insides of these two-dimensional cells can be considered as the quarter of a greater cell. This allows completing the structure of the pattern P into a finite three-dimensional cell which is colored in the functional areas layer as any three-dimensional cell that appears in the subshift.

- When the pattern intersects an infinite gray face, one can complete the pattern into the structure layer such that the projection over the copy of X_{adR} parallel to the face is a supertile. We use Proposition 3.28 in case that the patterns intersect the area between infinite two-dimensional

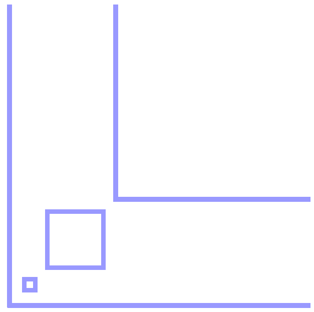


Figure 6.31: Illustration of the sequence of implications of the coherence rule of the functional areas sublayer.

supertiles over the face. We chose the order of the cell according to the value of the p-counter on the border of this cell. As we have coherence rules for the inside of the faces, this supertile can be colored as any finite supertile on the faces in the functional areas layer. There are the following possibilities (in all these cases, the symbols on this supertile are determined):

- the central corner of the supertile is colored in the functional areas sublayer, and as a consequence is colored with the (,) in the active areas sublayer. The color of the petal intersecting this one, and then the other petals in the supertile, is thus determined according to the orientation symbol written on the central corner.
- this corner is colored $(\,, \rightarrow)$ in the functional areas sublayer, and else (,) or (,) in the active areas sublayer.
- it is colored or $(\,, \rightarrow)$ in the functional areas sublayer, and else (,) or (,) in the active areas sublayer.
- this corner is colored with , and there is no restriction in the active areas sublayer.

Concerning the functional areas sublayer, all these cases can be encountered in any arbitrary large colored faces with arbitrary value of the counter. Indeed, they are encountered on any supertile whose orientations with respect to its $n+1$, $n+2$ and $n+3$ supertiles lie in a particular set. This set corresponds to the rules of the functional areas sublayer, and to the condition that the $n+3$ order supertile is colored blue. This happens for all the values of the p-counter in any two-dimensional cell. Because some columns or lines are active and others not, all the cases in the active areas sublayer are possible, with restrictions corresponding to the coherence rules.

- The pattern intersects an edge between gray faces. This case is similar to the previous one, since the coherence rules impose that the symbols have to match on the sides of the edges.

6.5.2 Completion of the linear counter and machines computations

In this section, we tell how to complete the pattern over this supertile in the linear counter and machines layers. Since there are rules connecting the symbols of the two layers on the edges, we only have to tell how to complete separately these two layers when knowing only one part (near a corner, a edge, or inside) of the face corresponding to else the linear counter or the machine.

The linear counter:

1. when knowing a part of face $1, 2, 3, 4$ or 5 which does not intersect the bottom line of face 1, top line of faces $4, 5$ and right column for faces 2 (meaning places where there is incrementation of the counter), the only difficulty for completing comes from the freezing signal, which is only when all the letters are equal to $c_{\mathtt{max}}$. Thus when the supertile is colored with this signal, there is nothing to do but to complete the face adding only letters equal to $c_{\mathtt{max}}$. When the freezing signal is , then we have to add letters that are different from $c_{\mathtt{max}}$.

2. when knowing some part of the incrementation positions, the completion depends on if we known the right or left part of the line (looking in the direction of face 1):

   - when knowing the right part, the freezing signal on the top face and bottom face are determined. We complete the left part of the bottom line of face 1 according to the freezing signal on the right part. If at some point the freezing signal becomes after being (from left to right), this means that we can complete the letters on the right equal to $c_{\mathtt{max}}$. When the freezing signal is all blank, we can complete them by adding at least some letter different from $c_{\mathtt{max}}$, and when it is all , we can complete them by adding only letters equal to $c_{\mathtt{max}}$.

The machines. When completing the machine face, there are two types of difficulties. The first one is managing the various signals : machine signals, first error, empty tape, empty side signals, and error signals. The second one is managing the space-time diagram of the machine. When the machine face is all known, there is no completion to make. Hence we describe the completion according to the parts of the face that are known (meaning that they appear in the initial pattern). Since there is strictly less difficulty to complete knowing only a part inside the face than on the border (since the difficulties come from completing the space-time diagram of the machine, in a similar way than for the border), we describe the completion only when a part of the border is known.

- When knowing the top right corner of the machine face:

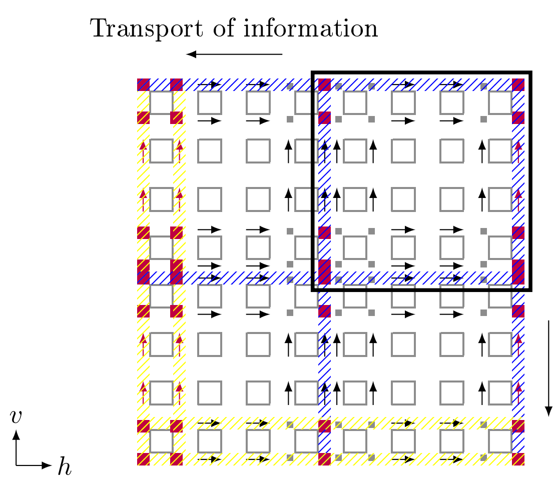


Figure 6.32: Illustration of the completion of the `on/off` signals and the space-time diagram of the machine. The known part is surrounded by a black square.

   1. if the signal detects the first machine in error state from left to right (becomes after being), then we already know the propagation direction of the error signal. Then we complete

the first machine signal and the error signal according to what is known. For completing the space-time diagram of the machine, the difficulty comes from the fact that this is possible that when completing the trajectory of two machine heads according to the local rules, they have to collide reversely in time. This is not possible in our model. This is where we use the `on`/`off` signals. We complete first the non already determined signals using only the symbol `off`, as illustrated on Figure 6.32.

Then the space-time diagram is completed by only transporting the information.

In the end, we completed with any symbols in $\mathcal{Q}$ or $\mathcal{A} \times \mathcal{Q}$ where the symbols are not determined. We can do this since they do not interact with the known part of the space-time diagram. Then we complete empty tape and empty side signals according to the determined symbols.

2. if the signal is all , then one can complete the face without wondering about the error signal.
3. if the signal is all , we complete in the same way as for the first point, and the `off` signals contribute to the first machine signal being all . If there is no error signal on the right side, one can complete so that the first machine in error state has above the arrow $\leftarrow$ indicating that the error signal has to propagate to the left. If there is an error signal on the right, then we set this arrow as $\rightarrow$. This is illustrated on Figure 6.33.

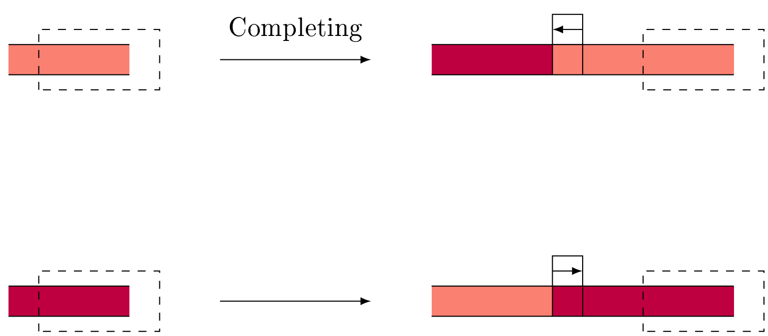


Figure 6.33: Illustration of the completion of the arrows according to the error signal in the known part of the area, designated by a dashed rectangle.

- when knowing the top left corner, the difficulty comes from the direction of propagation of the error signal. This is ruled in a similar way as point 3 of the last case.
- when knowing the bottom right corner or bottom left corner: the completion is similar as in the last points, except that we have to manage the empty tape and sides signals. The difficult point comes from the signal, whose propagation direction is towards the known corner. If this signal detects an error before entering in the known area, we complete so that the added symbols in $\{\texttt{on}, \texttt{off}\}$ are all `off`: this induces the error. When this signal enters without detecting an error, we complete all the symbols so that they do not introduce any error. A particular difficulty comes from the case when the bottom right corner is known. Indeed, when the signal enters without having detecting an error, this means we have to complete the pattern so that a machine head in initial state is initialized in the leftmost position of the bottom row. Since the pattern is completed in the structure layer into a great enough cell, this head can not enter in the known pattern.

- when the pattern crosses only a edge or the center of the face, the completion is similar (but easier since these parts have less information, then we need to add less to the pattern).

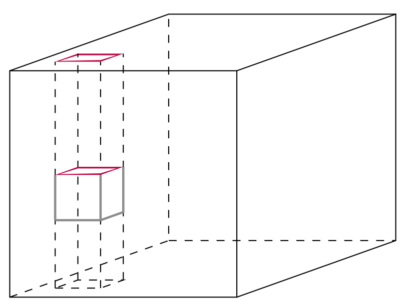
What is left to describe for the completion into a cubic supertile is the completion of the hierarchical counter layer.

6.5.2.1 Completion of the hierarchical counter layer

We can complete the three copies of the hierarchy bits and random bits on the completed supertile in a coherent way with the hierarchy bits and random bits on the pattern P. The colors in this layer are determined by a triple of colors in $\{ , \}$, corresponding to petal having maximal order in the three supertiles defining the cubic supertile. We have to prove how to view a three-dimensional cell having order $n+1$ with these colors in a greater three-dimensional cell, since as a consequence we can see the supertile around this order $n+1$ cell in the orientation corresponding to orientation marked on the supertile. It is sufficient to view this order $n+1$ three-dimensional cell as a part of an order $2^k p - 1$ cell with k great enough that is inside a greater cell and nearest to one of the corner (the colors of its borders are , ,). There are three cases:

1. When the order $n+1$ cell has its three colors equal to , then it can be found near the extremal corners of the $2^k p - 1$ one (thanks to the transformation rules of the hierarchy bits layer).

2. When there is only one color equal to , then consider the set of cells near a edge, having its direction orthogonal to the directions of the face of the three-dimensional cell having border colored . This corresponding two-dimensional cell is colored for all these three-dimensional cells. We pick one of the cells in this set such that the other faces have border colored with . This is possible because the other two two-dimensional cells have the same color for the cells in this set (because the $2^k p - 1$ order cell has its borders all colored with).



3. When there are two colors equal to , then we pick some order order $n+1$ two-dimensional cell on on the diagonal of the face of the $2^k p - 1$ order three-dimensional cell parallel to the face of the $n+1$ order cell which border is colored , such that the line and columns of order $n+1$ two-dimensional cells on the face contains two-dimensional cells colored with (for instance the middle lines/columns of the face). Hence we can pick a cell with colors (, ,) in the column of order $n+1$ three-dimensional cell under the picked two-dimensional cell.

6.5.3 Computation of the entropy dimension

In this section we prove a formula for the entropy dimension of the subshift X_z:

Proposition 6.13. *For all $z > 0$ a Δ_2-computable number, the entropy dimension of X_z is:*

$$\boxed{D_h(X_z) = \frac{1}{p} + z\left(1 - \frac{1}{2p}\right)}$$

Proof. Let us give upper and lower bounds on the limsup and liminf of

$$\frac{\log_2(\log_2(N_n(X_z)))}{\log_2(n)}.$$

Let P some n-block in the language of X_z.

1. **Upper bound:**

   One can complete P into an an order $2pq_n + 2$ supertile, where

   $$q_n = \lceil \log_2(n)/2p \rceil + 1.$$

   The number of patterns over some k order supertile is less than K times the number of random bits layout over a k supertile times the number of possible values of the linear counter inside a supertile, where $K > 0$ is a constant (indeed, all the other layers are determined by a symbol in a finite set).

   As a consequence of Lemma 6.11, and Lemma 6.7, the number of patterns over a $2pq_n + 2$-order supertile is smaller than

   $$K.|\mathcal{A}_c|^{2^{p+1}.(p+1)^{q_n}}.2^{\alpha(p).{q_n}^{\lambda(p)}}.4^{c^0_{q_n}}.16^{pc^1_{q_n}}.$$

   This implies that

   $$\log_2(N_n(X_z)) \leq \log_2(K) + 2^{p+1}.(p+1)^{q_n}\log_2(|\mathcal{A}_c|) + \alpha(p).{q_n}^{\lambda(p)}.4^{c^0_{q_n}}.16^{pc^1_{q_n}},$$

   and then

   $$\begin{aligned}\frac{\log_2 \circ \log_2(N_n(X_z))}{\log_2(n)} \quad \leq \quad & \frac{\log_2(\alpha(p).{q_n}^{\lambda(p)})}{\log_2(n)} + 2\frac{c^0_{q_n}}{\log_2(n)} + 4\frac{pc^1_{q_n}}{\log_2(n)} \\ & + \frac{1}{\log_2(n)}\log_2\left(1 + \frac{\log_2(K) + 2^{p+1}.(p+1)^{q_n}\log_2(|\mathcal{A}_c|)}{2^{pc^1_{q_n}}}\right)\end{aligned}$$

   The first term of this sum tends towards zero, since

   $$\frac{\log_2(\alpha(p).{q_n}^{\lambda(p)})}{\log_2(n)} = O\left(\frac{\log_2 \circ \log_2(n)}{\log_2(n)}\right)$$

   The fact that the third term tends towards zero comes from the choice of p. Indeed, we have

   $$\log_2(K) + 2^{p+1}.(p+1)^{q_n}\log_2(|\mathcal{A}_c|) = O(2^{\log_2(n)\log_2(p+1)/2p}) = O(n^{m/2p}).$$

   Moreover, by definition

   $$\frac{c^1_{q_n}}{q_n} \to z/2$$

   and as a consequence

   $$2^{pc^1_{q_n}} \geq n^{z/2}$$

   Since $m/2p < z/2$, this means the third term tends towards zero. Thus we have

   $$\limsup_n \frac{\log_2 \circ \log_2(N_n(X_z))}{\log_2(n)} \leq \limsup_n 2\frac{c^0_{q_n}}{\log_2(n)} + 4\frac{pc^1_{q_n}}{\log_2(n)} \leq 2\frac{(1-z/2)}{2p} + 4p\frac{z/2}{2p} = 1/p + z(1 - 1/2p)$$

2. **Lower bound:**

   For the lower bound, we do the reverse inclusion: if m is chosen great, any n-block P in the language of the subshift X_z contains some order $2pq'_n + 2$ supertile,

   $$q'_n = \lfloor \log_2(n)/2p \rfloor - 2,$$

   since these supertiles are repeated with period $2^{2pq'_n+4} \leq \frac{n}{2^{4p-4}}$, and have size $2^{2pq'_n+3} \leq \frac{n}{2^{4p-3}}$. Then the number of n-blocks in the language of X_z is greater than

   $$2^{4^{c^0_{q'_n}} . 16^{pc^1_{q'_n}}}.$$

   Similar computation as done for the upper bound leads to

   $$\liminf_n \frac{\log_2 \circ \log_2(N_n(X_z))}{\log_2(n)} \geq 1/p + z(1 - 1/2p),$$

   which means X_z has an entropy dimension and this dimension is equal to

   $$D_h(X_z) = 1/p + z(1 - 1/2p).$$

□

6.5.4 Minimality

In this section we prove that the subshift X_z is minimal. See 6.34 for a schema of the proof. This proof relies on the following lemma:

Lemma 6.14 (Globach's theorem). *The numbers $F_n = 2^{2^n} + 1$, $n \geq 0$ are coprime.*

Proof. Let $m > n \geq 0$. Then

$$F_m = 2^{2^m} + 1 = (2^{2^n} + 1 - 1)^{2^{m-n}} + 1 = \sum_{k=0}^{2^{m-n}} \binom{2^{m-n}}{k} (-1)^{-k} F_n^k + 1$$

$$= F_n \sum_{k=1}^{2^{m-n}} \binom{2^{m-n}}{k} (-1)^k F_n^{k-1} + 2$$

This means that a common divisor of F_n and F_m divide 2, but 2 does not divide F_n, so F_n and F_m are coprime. □

Consider some block P in the language of X, and complete it into a pattern P' over an order $2^k p$ three-dimensional cell. Pick some configuration $x \in X$, and consider the restriction of x on the hierarchy bits layer (over the copy of the Robinson subshift parallel to $\mathbf{e}^2$ and $\mathbf{e}^3$). On can find some $\mathbf{u}$ such that the projection of the three dimensional cell over the hierarchy bits layer appear in position $\mathbf{u}$ (using the same arguments as in Section 6.5.2.1).

For all i, a three-dimensional cell appears on position $(2i4^{2^k p})\mathbf{e}^1 + \mathbf{u}$. Comparing the cell in position $(v + 2i4^{2^k p})\mathbf{e}^1 + \mathbf{u}$ and $(v + 2(i+1)4^{2^k p})\mathbf{e}^1 + \mathbf{u}$, the second one has the same linear counters as the first one, and the jth hierarchical counters values are incremented $4^{(2^k - 2^j)p}$ times for all j in the the second one, with respect to the first.

Let t be the following application:

$$t \;:\; \begin{array}{ccc} \mathbb{Z}/p_1\mathbb{Z} \times ... \times \mathbb{Z}/p_k\mathbb{Z} & \rightarrow & \mathbb{Z}/p_1\mathbb{Z} \times ... \times \mathbb{Z}/p_k\mathbb{Z} \\ (i_1, ..., i_k) & \mapsto & (i_1 + 4^{(2^k-1)p}, ..., i_k + 1) \end{array},$$

where $p_1, \ldots, p_k$ are the periods of k first hierarchical counters. This is a minimal application, meaning that for all $\mathbf{i}, \mathbf{j} \in \mathbb{Z}/p_1\mathbb{Z} \times ... \times \mathbb{Z}/p_k\mathbb{Z}$, there exists some n such that $t^n(\mathbf{i}) = \mathbf{j}$. Indeed, considering some $\mathbf{i}$, denote n_0 the smallest positive integer such that $t^{n_0}(\mathbf{i}) = \mathbf{i}$. This means that p_j divides $n_0 4^{(2^k-2^j)p}$ for all j, and because p_j is a Fermat number, it is odd, and this implies that p_j divides n_0. For the numbers p_j are coprime (Lemma 6.14), this implies that $p_1 \times ... \times p_j$ divides n_0. Because this number is a period of the application t, this means that $p_1 \times ... \times p_j$ is the smallest period of every element of $\mathbb{Z}/p_1\mathbb{Z} \times ... \times \mathbb{Z}/p_k\mathbb{Z}$ under the application t.

As a consequence, for all $\mathbf{i}$, the finite sequence $(t^n(\mathbf{i}))$, for n going from 0 to $p_1 \times ... \times p_j - 1$, takes all the possible values in $\mathbb{Z}/p_1\mathbb{Z} \times ... \times \mathbb{Z}/p_k\mathbb{Z}$.

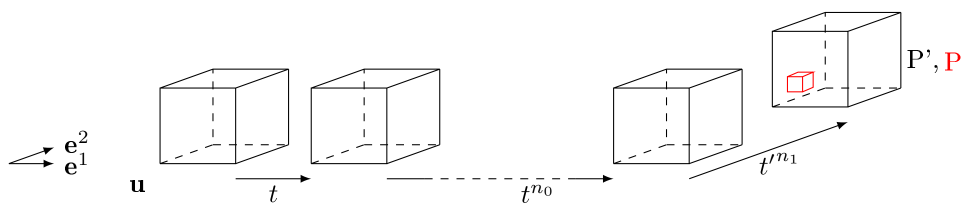


Figure 6.34: Schema of the proof for the minimality property of X_z.

Hence one can find in x an order $2^k p$ three-dimensional cell which supports values of the hierarchical counter equal to the ones that are in P', iterating n_0 times this shift.

Now consider $\mathbf{u}'$ the position where this cell appears in x, and the cells appearing in the same configuration on positions $2i4^{2^k p}\mathbf{e}^2 + \mathbf{u}'$. Using a similar argument as above, on the function

$$t' \;:\; \begin{array}{ccc} \mathbb{Z}/p'_1\mathbb{Z} \times ... \times \mathbb{Z}/p'_k\mathbb{Z} & \rightarrow & \mathbb{Z}/p'_1\mathbb{Z} \times ... \times \mathbb{Z}/p'_k\mathbb{Z} \\ (i_1, ..., i_k) & \mapsto & (i_1 + 4^{(2^k-1)p}, ..., i_k + 1) \end{array},$$

where $p'_1, \ldots, p'_k$ are the periods of k first linear counters, one can find some i such that on position $2i4^{2^k p}\mathbf{e}^2 + \mathbf{u}'$, the cell supports the same values of linear counters than P', iterating n_1 times this shift. Moreover, since the hierarchical counters are synchronized in the direction $\mathbf{e}^2$, this cells has also the same hierarchical counter values as P'.

For the pattern on the cell is determined by the values of the counters, this cell supports the pattern P'.

Hence X_z is a minimal SFT.

Remark 41. *It is trivial to find some minimal $\mathbb{Z}^3$-SFT having entropy dimension equal to zero. However, it is not to find one having entropy dimension equal to two, and we don't know a simpler proof than the construction presented in this text.*

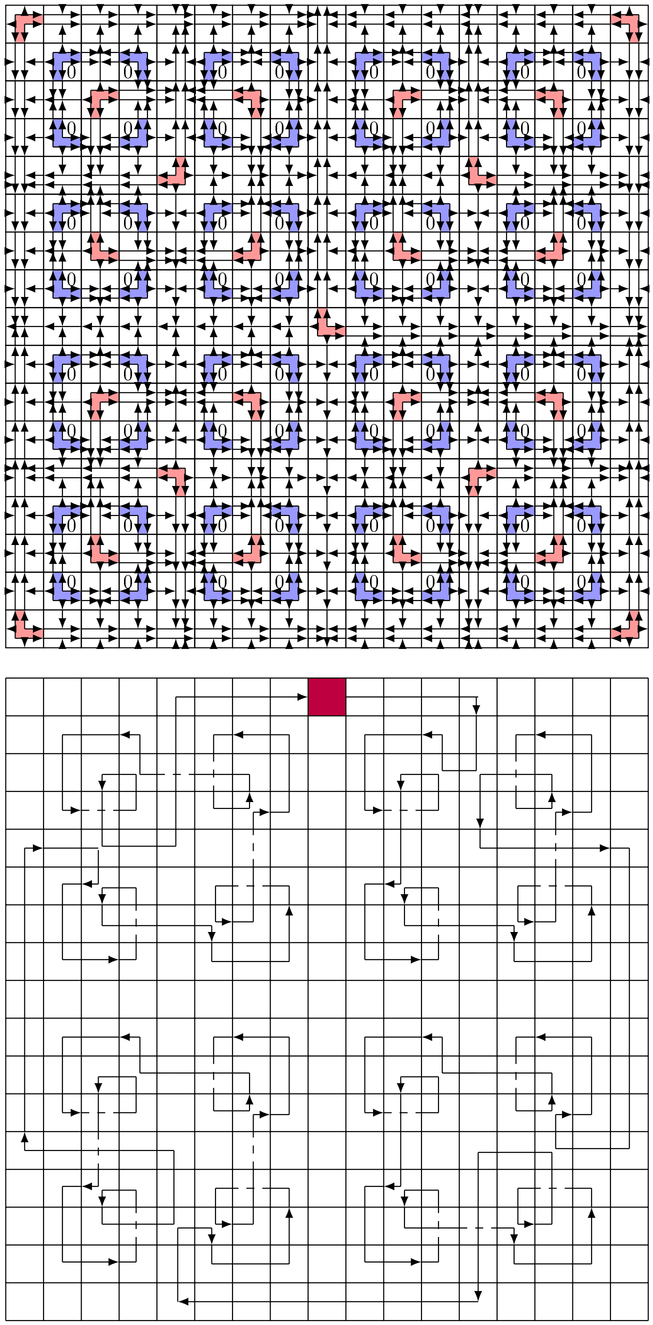


Figure 6.35: Illustration of the convolutions rules.

Chapter 7

Computable dynamical systems on computable metric spaces

Sommaire

The point of this last chapter is to extend the investigation about the computability of growth-type invariants, in particular under constraints, to more general dynamical systems on a compact metric space. For this purpose, we need a notion of computability adapted to these systems. This chapter is a joint work with Mathieu Sablik, Cristobal Rojas and Benjamin Hellouin.

7.1 Introduction

The theory of computation is well established over finite objects (as integers). This means objects having a representation as a finite layout of symbols in a finite set (like the binary representation for integers). On the other hand, real numbers are thought as infinite since they can not be represented in a finite way. A function f that has finite objects as arguments and values is said to be **computable** when it there is an algorithm that, taking as input the representation of the object x, produces a representation of the object $f(x)$.

The purpose of *computable analysis* is to understand to which extent some infinite objects are computable, that is to say that there exists a computing device able to produce a meaningful representation of it.

This chapter is organized as follows. In Section 7.2, after recalling notions of computable analysis on metric spaces, we provide a definition of computability for dynamical systems on compact metric spaces, and then characterizations of the definitions on the Cantor set and the compact interval. Most of the definitions and statements of this section are known. In Section 7.3, we recall definitions for the growth type invariants studied in the previous chapters for dynamical systems on compact metric spaces, and prove general computability obstructions on these invariants for computable dynamical systems, and obstructions for expansive systems.

In Section 7.5, we characterize the possible entropies of computable dynamical systems on the Cantor set, and give a partial realization under the constraint of surjectivity.

In Section 7.6, we characterize the possible entropies of computable dynamical systems on the interval, and characterize the entropies under constraint on the variation and constraint of expansivity.

We focus here on one-dimensional dynamical systems (that is to say that the dynamical system is a $\mathbb{Z}$-action), but the definitions and results can be adapted to higher dimensional ones ($\mathbb{Z}^d$-actions).

7.2 Computable analysis on computable compact metric spaces

In this section, we recall notions of computable analysis on metric spaces. All the results presented here are well known. For a reference on computable analysis, see [BHW08].

7.2.1 Open covers

In this section, we recall notions about open covers for a metric space.

Definition 7.1. *Let (X, d) be some metric space, and K a compact subset of X. A **cover** of K is a set $\mathcal{U}$ of open subsets such that*

$$K \subset \bigcup_{U \in \mathcal{U}} U.$$

Definition 7.2. *Let $\mathcal{U}$ and $\mathcal{U}'$ be two sets of open subsets of X. We say that $\mathcal{U}$ is **thinner** than $\mathcal{U}'$ when for all $U \in \mathcal{U}$, there is some $V \in \mathcal{U}'$ such that $U \subset V$. We denote this relation*

$$\mathcal{U} \prec \mathcal{U}'.$$

Definition 7.3. *Let $\vee$ be the operator on open covers such that for all pairs of open covers $\mathcal{U}$ and $\mathcal{U}'$,*

$$\mathcal{U} \vee \mathcal{U}' = \{U \cap U', U \in \mathcal{U},\ U' \in \mathcal{U}'\}.$$

Definition 7.4. *A **sub-cover** of some cover $\mathcal{U}$ is a subset of this set which is also a cover.*

Remark 42. *For a continuous function $X \to X$, K a compact subset of X, an open cover $\mathcal{U}$ of K, the set*

$$f^{-1}(\mathcal{U}) = \{f^{-1}(U) \ : \ U \in \mathcal{U}\}$$

is an open cover.

7.2.2 Computable metric spaces

Definition 7.5. *A **computable metric space** is some $(X, d, \mathcal{S})$, where (X, d) is a metric space, $\mathcal{S} = \{s_i : i \geq 0\}$ a countable dense subset of X, whose elements are called **ideal points**, such that there exists an algorithm which given as input (i, j, n) a triple of integers, outputs some r such that*

$$|d(s_i, s_j) - r| \leq 2^{-n}.$$

We say that the distances between ideal points are uniformly computable.

For $r > 0$ a rational number, and x an element of X, we denote $B(x, r) = \{z \in X \ : \ d(z, x) < r\}$, the ball centered on x with radius r. The balls centered on an element of $\mathcal{S}$ are called **ideal balls**. Consider some computable invertible map $\varphi : \mathbb{N} \to \mathbb{N} \times \mathbb{Q}$, fixed for the following. We denote $B_n = B(s_{\varphi_1(n)}, \varphi_2(n))$, where $\varphi(n) = (\varphi_1(n), \varphi_2(n))$. We will denote $s^{(n)} = s_{\varphi_1(n)}$ the center of the ball B_n, and $r^{(n)} = \varphi_2(n)$ its radius.

Moreover, for any subset (finite or infinite) of $\mathbb{N}$, we denote $\mathcal{U}_I$ the collection of the balls B_n, $n \in I$, and U_I the union of these balls:

$$U_I = \bigcup_{n \in I} B_n$$

Example 7.6.
- *For any $\mathcal{A}$ finite alphabet, and $\#$ one of its elements, the Cantor space $\mathcal{A}^{\mathbb{N}}$ with its usual metric and $\mathcal{S} = \{w\#^{\infty} : w \in \mathcal{A}^*\}$, is a computable metric space. In this case, the ideal balls are the cylinders. As well, any full shift on an alphabet $\mathcal{A}$ on $\mathbb{Z}^d$, $d \geq 1$ has a natural structure of computable metric space.*
- *The compact interval $[0, 1]$, with its usual metric and $\mathcal{S}$ the set of rational numbers in this interval, is a computable metric space. The ideal balls are all the open intervals with rational extreme points.*

Definition 7.7. *Let $(X, d, \mathcal{S})$ be a computable metric space, such that (X, d) is compact. It is said to be **recursively compact** if the inclusion*

$$X \subset U_I,$$

where I is a finite subset of $\mathbb{N}$, is semi-decidable. This means that there is an algorithm which given as input some I, stops if and only if this inclusion is verified.

Example 7.8. *The Cantor space and the compact interval are recursively compact.*

7.2.3 Computable closed subsets

Definition 7.9. *Let $(X, d, \mathcal{S})$ be a computable metric space. A closed subset $K \subset X$ is said to be **effective** when there exists an algorithm such that*

$$X \backslash K = U_I,$$

where I is the set of integers on which this algorithm stops.

Proposition 7.10. *Let (X, d) be a compact metric space. Let K be a closed subset of X. The following conditions are equivalent, for I finite subset of $\mathbb{N}$:*

1. $K \subset U_I$
2. *there exists J finite subset of $\mathbb{N}$ such that for all $n \in J$, $B_n \subset X \backslash K$ and $X \subset U_{I \cup J}$.*

Proof. Indeed, the first condition implies that, if J' is the set of the integers n such that $B_n \subset X \backslash K$, then $X \backslash K = U_{J'}$. This means that $X \subset U_{I \cup J'}$. Since X is a compact space, there exists J some finite subset of J' such that $X \subset U_{I \cup J}$. Reciprocally, if this condition is true, since for all $n \in J$, $B_n \subset X \backslash K$, this implies that $K \subset U_I$. □

Definition 7.11. *We say that two ideal balls B_n and B_k are **manifestly disjoint** when*

$$d(s^{(n)}, s^{(k)}) > r^{(n)} + r^{(k)}.$$

Proposition 7.12. *Let (X, d) be a compact metric space with a dense subset $\mathcal{S}$, and K a closed subset of X. We have the following equality:*

$$X \backslash K = U_I,$$

where I is the set of integers n such that there exists some finite set J of integers such that for all $k \in J$, B_n and B_k are manifestly disjoint, and

$$K \subset U_J.$$

Proof.

1. Let $x \in X \backslash K$. Let us denote $d(x, K) = \min_{y \in K} d(x, y)$ the distance between the point x and the closed set K. Since this point is not in K, and K is a compact space, $d(x, K) > 0$. There exists some J such that

   $$K \subset U_J$$

   and that the radius of all the ball B_n, $n \in J$, is $< \frac{1}{2} d(x, K)$. Moreover, there exists some $r < \frac{1}{2} d(x, K)$ such that

   $$B(x, r) \subset X \backslash K.$$

   This ball contains some B_k such that $x \in B_k$. This is illustrated on Figure 7.1.

   As a consequence, for all $n \in J$,

   $$r^{(n)} < \frac{1}{2} d(x, K) \leq d(x, K) - r^{(k)} \leq d(s^{(n)}, s^{(k)}),$$

   since

   $$d(s^{(n)}, s^{(k)}) \geq d(x, s^{(n)}) - d(x, s^{(k)}) \geq d(x, K) - r^{(k)},$$

   since the ball B_k contains x.

2. Reciprocally, if x is contained is some ball B_k and there exists some J such that

   $$K \subset \bigcup_{n \in J} B_n,$$

   and for all $n \in J$,

   $$d(s^{(n)}, s^{(k)}) > r^{(n)} + r^{(k)},$$

   then this inequality ensures that $B_n \bigcap B_k = \emptyset$ for all $n \in J$. As a consequence $x \in X \backslash K$.

□

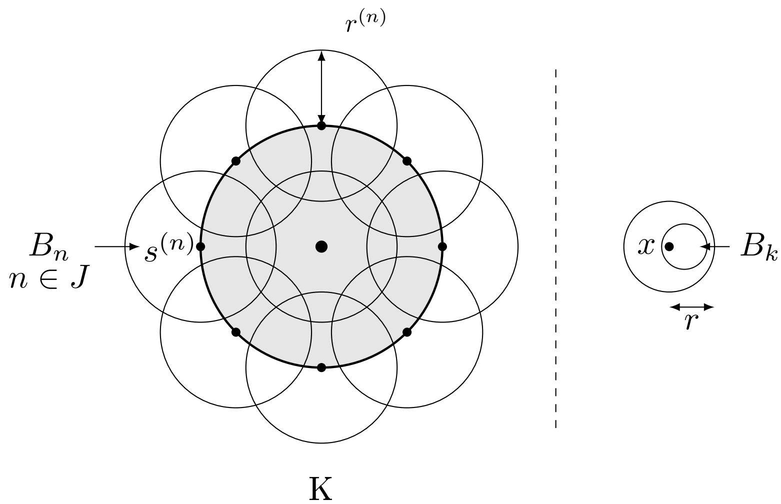


Figure 7.1: Illustration of the proof of Proposition 7.12

Proposition 7.13. *Let $(X, d, \mathcal{S})$ be a recursively compact metric space. A closed subset $K \subset X$ is a effective if and only if the inclusion*

$$K \subset U_I,$$

where I is a finite subset of $\mathbb{N}$, is semi-decidable.

Proof.
- Assume that K is effective. The inclusion $K \subset U_I$ is semi-decided by testing for all the finite subsets J of $\mathbb{N}$, if $X \subset U_{I \cup J}$. Since $(X, d, \mathcal{S})$ is recursively compact, and by Proposition 7.10, this algorithm ends if the inclusion $K \subset U_I$ is true.
- Reciprocally, if the inclusion

$$K \subset U_I$$

can be semi-decided, since the distances between ideal points are uniformly computable, the conditions

$$K \subset U_J$$

and $d(s^{(n)}, s^{(k)}) > r^{(n)} + r^{(k)}$ can be semi-decided. This means, by Proposition 7.12, that K is effective.

□

Remark 43. *In particular, we recover that a subshift of the full shift $\mathcal{A}^{\mathbb{N}}$ is effective if and only if it can be generated by an effective language.*

Proposition 7.14. *Let $(X, d, \mathcal{S})$ be a recursively compact metric space. The inclusion $\overline{B_n} \subset B_m$ is semi-decidable.*

Proof. For all n, the set

$$X \backslash \overline{B_n} = U_{J_n},$$

where J_n is the set of k such that the balls B_k and B_n are manifestly disjoint. Since this inequality is semi-decidable, the closed subset $\overline{B_n}$ is effective. From Proposition 7.13, we deduce that the inclusion

$$\overline{B_n} \subset U_{J_n}$$

for a finite set J is semi-decidable, and in particular, the inclusion $\overline{B_n} \subset B_m$ is semi-decidable. □

7.2.4 Computable functions

Definition 7.15. *Let (X, d) be some compact metric space. A function $f : X \to X$ is **computable** when there exists an algorithm which given as input some integer m, enumerates a set I_m such that*

$$f^{-1}(B_m) = U_{I_m}.$$

Remark 44. *As a consequence of this definition, a computable function is continuous.*

For X a compact computable metric space we consider the space $X \times X$ as a metric space with the distance d^2 defined for all $(x, y), (x', y') \in X \times X$ by

$$d^2((x, y), (x', y')) = \max(d(x, y), d(x', y')).$$

The space $(X^2, d^2, \mathcal{S}^2)$ is a compact computable metric space. The ideal balls of this space are the products of ideal balls of X.

Proposition 7.16. *Let (X, d) be some compact computable metric space. The distance function $d : X \times X \to [0, \delta(X)]$ is computable.*

Proof. It is sufficient to see that the conditions

$$d(s^{(n)}, s^{(k)}) - r^{(n)} - r^{(k)} > l$$

and

$$d(s^{(n)}, s^{(k)}) + r^{(n)} + r^{(k)} < r$$

for s, n integers and r, r_n, r_k rational numbers, are semi-decidable, since any point in $d^{-1}(]l, r[)$ is included in some $B_n \times B_k$ such these inequalities are verified. □

Example 7.17. *In particular, a computable function $f : \mathcal{A}^{\mathbb{N}} \to \mathcal{A}^{\mathbb{N}}$ is a function such that there exists some non decreasing computable function $\varphi : \mathbb{N} \to \mathbb{N}$, and an algorithm which gives the nth first symbols of the sequence $f(x)$ from input of the $\varphi(n)$ first symbols of the sequence x.*

In general, we have the following:

Proposition 7.18. *Let $(X, d, \mathcal{S})$ be a recursively compact metric space. If $f : X \to X$ is a computable function, then there exists an algorithm which on input (k, l), outputs some number n such that*

$$d(f(s_k), s_n) \leq 2^{-l}.$$

Proof. The algorithm works as follows:

1. It tests for all the finite sets I such that for all $m \in I$, the mth ideal ball has radius 2^{-l}, until it finds one such that
$$X \subset U_I.$$
This test uses the fact that X is recursively compact.
2. For this I, and for all $m \in I$, it enumerates a set I_m such that
$$f^{-1}(B_m) = U_{I_m}.$$
3. For all the elements p of the sets I_m, and q such that the ball B_q is centered on s_k, it tests if
$$\overline{B_q} \subset B_p.$$
If this inclusion is true, then the algorithm outputs the integer n such that the ball B_m is centered on s_n.

This algorithm stops since the inclusion of the last point is semi-decidable, and that there exists p, q such it is true. Moreover, by definition, it outputs n such that $d(f(s_k), s_n) \leq 2^{-l}$. See a schema on Figure 7.2.

□

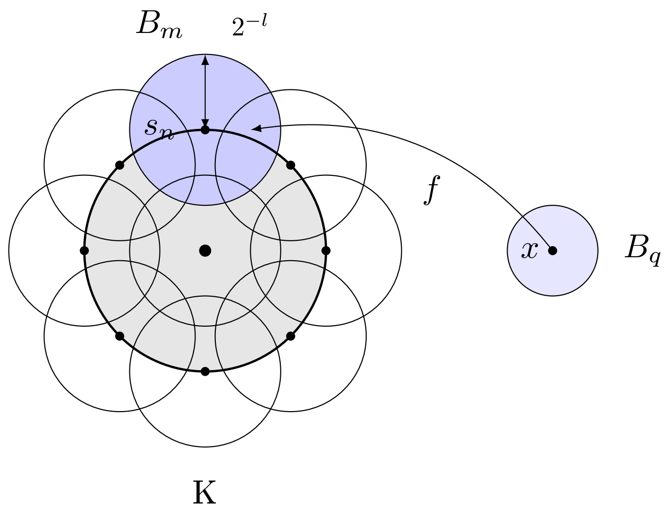


Figure 7.2: Schema of the proof of Proposition 7.18

7.3 Computability of entropy and entropy dimensions of dynamical systems

In this section we recall the definitions of some growth-type invariants, namely the entropy and entropy dimensions, for topological dynamical systems. Then, we provide an obstruction on the computability of these invariants for computable dynamical systems on a recursively compact metric space. We consider the effect of expansiveness constraint on the computability of these invariants.

In all the following, we use the following definition of a dynamical system:

Definition 7.19. *Let (X,d) be a compact metric space. A **dynamical system** of the ambient space X is some (K,f), where K is a compact subset of X, and $f : X \to X$ a continuous function.*

In all this section, we fix (X,d) some compact metric space.

7.3.1 Entropy

We define here the entropy and elementary properties of this invariant. For the proofs of these properties, see [AKM65].

Definition 7.20. *The **entropy** of a dynamical system (K,f) of X **relative to an open cover** $\mathcal{U}$ is defined by*

$$h(K,f,\mathcal{U}) = \inf_n \frac{\log_2 \left(N_n \left(K,f,\mathcal{U}\right)\right)}{n}$$

where $N_n(K,f,\mathcal{U})$ is the minimal cardinality of a sub-cover of

$$\bigvee_{k \leq n-1} f^{-k}(\mathcal{U}).$$

Remark 45. *It is a well known fact that this infimum is also a limit. This fact relies on the sub-additivity of the sequence $(N_n(K,f,\mathcal{U}))_{n\in\mathbb{N}}$.*

Lemma 7.21. *If $\mathcal{U}$ and $\mathcal{U}'$ are two open covers such that $\mathcal{U} \prec \mathcal{U}'$, then $h(K,f,\mathcal{U}) \geq h(K,f,\mathcal{U}')$.*

Definition 7.22. *The **topological entropy** of a dynamical system (K,f) is defined as*

$$h(K,f) = \sup \{h(K,f,\mathcal{U}) \mid \mathcal{U} \text{ finite open cover of } K\}.$$

Proposition 7.23. *The topological entropy is invariant under conjugacy.*

Lemma 7.24. *For all $n \geq 1$, and (K,f) a dynamical system in X, $h(f^n) = nh(f)$.*

7.3.2 Entropy dimensions

A general definition of entropy dimensions was introduced in [DHKP11]. Proofs of their elementary properties can be found there, for instance that they are topological invariants. We recall their definition in this section.

The upper and lower entropy dimensions of a dynamical system (K,f) relative to an open cover $\mathcal{U}$ are defined respectively by

$$\overline{D_h}(K,f,\mathcal{U}) = \limsup_n \frac{\log_2 \circ \log_2 (N_n(K,f,\mathcal{U}))}{\log_2(n)}$$

and

$$\underline{D_h}(K,f,\mathcal{U}) = \liminf_n \frac{\log_2 \circ \log_2 (N_n(K,f,\mathcal{U}))}{\log_2(n)}.$$

Lemma 7.25. *If $\mathcal{U}$ and $\mathcal{U}'$ are two open covers such that $\mathcal{U} \prec \mathcal{U}'$, then $\overline{D_h}(K,f,\mathcal{U}) \geq D_h(K,f,\mathcal{U}')$, and $\underline{D_h}(K,f,\mathcal{U}) \geq D_h(K,f,\mathcal{U}')$.*

The upper and lower entropy dimensions are the respective numbers

$$\overline{D_h}(K,f) = \sup_{\mathcal{U}} \overline{D_h}(K,f,\mathcal{U}),$$

and

$$\underline{D_h}(K,f) = \sup_{\mathcal{U}} \underline{D_h}(K,f,\mathcal{U}).$$

When these two last numbers are equal, we say that the system has **entropy dimension**

$$D_h(f,K) = \overline{D_h}(K,f) = \underline{D_h}(K,f).$$

Proposition 7.26. *The entropy dimensions are invariants under conjugacy.*

7.4 Obstructions on the computability of entropy and entropy dimensions

In this section we provide general obstructions for the entropy and entropy dimensions of computable dynamical systems. Although stated for these two invariants, this proof can be extended to other growth-type invariants. We use the following notion of computability for a dynamical system:

Definition 7.27. *Let $(X,d,\mathcal{S})$ be a computable metric space. A dynamical system (K,f) is said to be computable when K is effective and f computable.*

7.4.1 The topological entropy is a Σ_2 computable number

The aim of this section is to prove the following theorem:

Theorem 7.28. *Let $(X, d, \mathcal{S})$ be a recursively compact metric space, and (K, f) a computable dynamical system in X, $h(K, f)$ is Σ_2-computable, $\overline{D_h}$ is Σ_3-computable, and $\underline{D_h}$ is Σ_2-computable.*

Its proof relies on the following intermediate statement:

Proposition 7.29. *Let $(X, d, \mathcal{S})$ be a recursively compact metric space, and (K, f) be a computable dynamical system in X. The sequence*

$$(N_n(K, f, \mathcal{U}_I))_{I,n}$$

is Π_1-computable.

Idea: *the idea of the proof is to construct an algorithm that, on input I, n, test successively for all the subsets of the cover $\bigvee_{k \leq n-1} f^{-k}(\mathcal{U}_I)$, if this subset is a subcover, and each time this algorithm detects a sub-cover, it registers in a variable the minimum of the current value of this variable and the cardinality of the sub-cover. This way it will print a sequence of integers whose infimum is $(N_n(K, f, \mathcal{U}_I))_{I,n}$.*

Proof. Let (K, f) be a computable dynamical system in X. Consider the algorithm, which from input (I, n), where I is a finite set of integers, and n an integer, works as follows:

1. it initializes a variable c with the value $c = +\infty$.
2. for all finite set of integers J, and L a subset of $[\![1, |I|]\!]^n$, it does the following operations:
   (a) it tests if $\mathcal{U}_I$ and $\mathcal{U}_J$ are covers of K ; this test uses the fact that X is recursively compact.
   (b) in parallel, it tests for all $k \in J$ if there exists some element $(l_1, ... l_n)$ of L such that

   $$\overline{B_k} \subset \bigcap_{i \leq n-1} U_{m_{l_i}},$$

   where m_{l_i} is the l_ith element of I. This means that $\mathcal{U}_J$ is a thinner cover than the subset of

   $$\bigvee_{k \leq n-1} f^{-k}(\mathcal{U}_I)$$

   corresponding to the lists of indexes in L.
   (c) if all these tests are true, then register in the variable c the value $\min(|L|, c)$.

Then consider the algorithm which, on input I, n, k executes the k first steps of the algorithm above, and outputs the current value of the variable c after these k steps. Denote $\xi : \mathbb{N} \to \mathbb{N}$ the computable function defined by this second algorithm.

We have, by definition, the equality

$$(N_n(K, f, \mathcal{U}_I))_{I,n} = \inf_k \xi(I, n, k).$$

□

Proof. of Theorem 7.28:

- **A formula for the entropy using ideal open covers:**

  Let (K,f) be a dynamical system in X. The supremum in the definition of the topological entropy can be taken over finite open covers of K with ideal balls.

  Indeed, for any cover $\mathcal{U}$ of K, $U \in \mathcal{U}$ and x an element of U, there exists an ideal ball $B_{n(x)} \subset U$ that contains x. The cover $\{B_{n(x)}, x \in K\}$ admits as sub-cover some $\mathcal{U}_I$ for I some finite set of integers, which is by construction thinner than $\mathcal{U}$. Thus,

  $$h(K,f,\mathcal{U}) \leq h(K,f,\mathcal{U}_I).$$

  As a consequence,

  $$h(K,f) = \sup_I h(K,f,\mathcal{U}_I),$$

  where the supremum is over all the finite set of integers.

- From the first point,

  $$h(K,f) = \sup_I h(K,f,\mathcal{U}_I),$$

  from Proposition 7.29, and the fact that the logarithm is a computable function, we get that $h(K,f)$ is a Σ_2 computable number. The obstructions for the entropy dimensions are straightforward, with similar arguments.

□

7.4.2 Constraint on the existence of a generating partition:

Definition 7.30. *Let (X,d) be a compact metric space, $f : X \to X$ a continuous function, and $\mathcal{U}$ a finite open cover of X. We say that $\mathcal{U}$ is a generating cover X when for all $\mathcal{U}'$ finite open cover of X, there exists some n such that the cover*

$$\bigvee_{k \leq n-1} f^{-k}(\mathcal{U})$$

is thinner than $\mathcal{U}'$.

Example 7.31. *We say that a dynamical system is expansive when there exists $\alpha > 0$ such that for any ordered pair of points $(x,y) \in X^2$, there exists some n such that*

$$d(f^n(x), f^n(y)) \geq \alpha.$$

It is a well known fact that an expansive dynamical system admits a generating cover. In this case, the entropy of this system is equal to the entropy relative to this cover.

Proposition 7.32. *Let $(X,d,\mathcal{S})$ be a recursively compact metric space, and (K,f) a computable dynamical system in X. If (K,f) has a generating cover, its entropy is a Π_1-computable number. Its upper (resp. lower) entropy dimension is a Σ_2 (resp. Π_2) computable number.*

Remark 46. *In particular, since any subshift is expansive, any effective subshift verifies this statement.*

Proof. This comes from the fact that when the system admits a generating cover, then the supremum over the covers is suppressed in the definitions. □

7.5 Dynamical systems on the Cantor set

In this section, we consider the computability of the entropy when the metric space is $X = \{0,1\}^{\mathbb{N}}$.

We first characterize in Section 7.5.1 the entropies of computable dynamical system in X. In Section 7.5.2 we consider the constraint of surjectivity.

7.5.1 Characterization of the entropies of computable dynamical systems on the Cantor set

The proof of this characterization is based on effective dynamical systems:

Definition 7.33. *An effective dynamical system (EDS) is a subset of $\{0,1\}^{\mathbb{N}\times\mathbb{Z}}$, defined by a recursively enumerable set of forbidden patterns. The ordered pair (X, σ^{e^2}) is a dynamical system.*

It is known that the numbers that are the entropy of an EDS are all the Σ_2-computable numbers:

Theorem 7.34 ([Hoc09]). *Let $h \geq 0$ be a Σ_2-computable real number. There exists an effective dynamical system $X \subset \{0,1\}^{\mathbb{N}\times\mathbb{N}}$ which entropy is h.*

We deduce from this result a characterization of the numbers that are the entropy of a computable dynamical system on $\{0,1\}^{\mathbb{N}}$:

Proposition 7.35. *Every Σ_2-computable non negative real number is the entropy of a computable function on the Cantor set.*

Remark 47. *The proof of Theorem 7.34 can be adapted in order to realize every Δ_2-computable number as the entropy dimension of an EDS. As a consequence, any Δ_2-computable number is the entropy dimension of a computable function on the Cantor set. The proof is similar to the proof of Proposition 7.35.*

Idea of the proof: *we recode the system obtained by Theorem 7.34 into a computable function on a closed subset X of the Cantor set into itself. Then we complete this function into a function on the whole Cantor set such that the trajectories of the points outside of the closed subset are attracted to it, using an error detection mechanism, which detects if a point is not in the closed subset.*

Proof. Let h be a Σ_2-computable real number, and X' an EDS whose entropy is h. Then consider the image of X' by the canonical injection

$$\{0,1\}^{\mathbb{N}\times\mathbb{Z}} \to \{0,1,\#\}^{\mathbb{N}\times\mathbb{Z}}.$$

1. **Recoding function:**

   Consider the computable function $\phi : \mathbb{N}\times\mathbb{Z} \to \mathbb{N}$ such that for all n, k, $\phi(2^{2n}-1+2^{2n+1}k) = (n,k)$ and $\phi(2^{2n+1}-1+2^{2n+1}k) = (n,-k-1)$. Consider also the function $\psi : \{0,1,\#\}^{\mathbb{N}\times\mathbb{Z}} \to \{0,1,\#\}^{\mathbb{N}}$ defined for all u, $n \geq 0$, and $k \geq 0$, by

$$\begin{array}{rcl} (\psi(u))_{2^{2n}-1+2^{2n+1}k} & = & u_{n,k} \\ (\psi(u))_{2^{2n+1}-1+2^{2n+1}k} & = & u_{n,-k-1} \end{array}$$

   Informally, the sequence $\psi(u)$ is obtained by writing on some n-tape the elements of the sequence $(u_{0,k})_{k\geq 0}$ every two positions, starting from the position 0, then on the positions that are left empty write the elements of the sequence $(u_{0,-k-1})_{k\geq 0}$ on every two position starting from position 1, then write the sequence $(u_{1,k})_{k\geq 0}$ in the same way on the positions that are left empty on every two position starting from 3, etc. This function serves to re-code the EDS as a computable dynamical system on a Cantor set.

2. **Completing the EDS into a computable function using error detection:**

   Let us denote Δ_m the set of positions $(n,k) \in \mathbb{N}\times\mathbb{Z}$, with $((k=-m$ or $k=m)$ and $n \leq m)$ or $(-m \leq n \leq m$ and $k=m)$. See Figure 7.3 for an illustration.

   Then we consider the computable function $f : \{0,1,\#\}^{\mathbb{N}\times\mathbb{Z}} \to \{0,1,\#\}^{\mathbb{N}\times\mathbb{Z}}$ defined as follows on a configuration x. For all $m \geq 0$, in order to determine the coefficients on positions in the set $\Delta_{m+1}\backslash\Delta_m$, consider the coefficients of the configuration x on $\Delta_{2(m+1)}$:

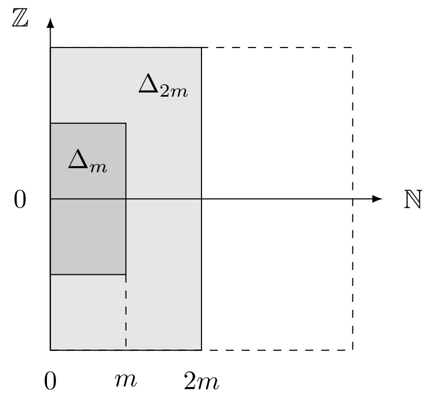


Figure 7.3: Illustration of the definition of the sets Δ_m and Δ_{2m}.

(a) if there is one equal to $\#$, or if the pattern $x_{\Delta_{2(m+1)}}$ contains some forbidden pattern amongst the first $m+1$ forbidden patterns of the EDS, then the coefficients of $f(x)$ on $\Delta_{m+1} \backslash \Delta_m$ are all equal to $\#$.

(b) in the other case, for all $\mathbf{v}$ in $\Delta_{m+1} \backslash \Delta_m$, $f(x)_{\mathbf{v}} = x_{\mathbf{v}+\mathbf{e}^2}$.

The symbol $\#$ is thought as an *error symbol*: when a forbidden pattern is detected, the function write a symbol $\#$ which then propagates on $\mathbb{N} \times \mathbb{Z}$: when x contains some symbol $\#$ on some Δ_m, then on every position outside of $\Delta_{\lceil m/2 \rceil}$, the symbol of $f(x)$ on this position is $\#$. See an illustration on Figure 7.4.

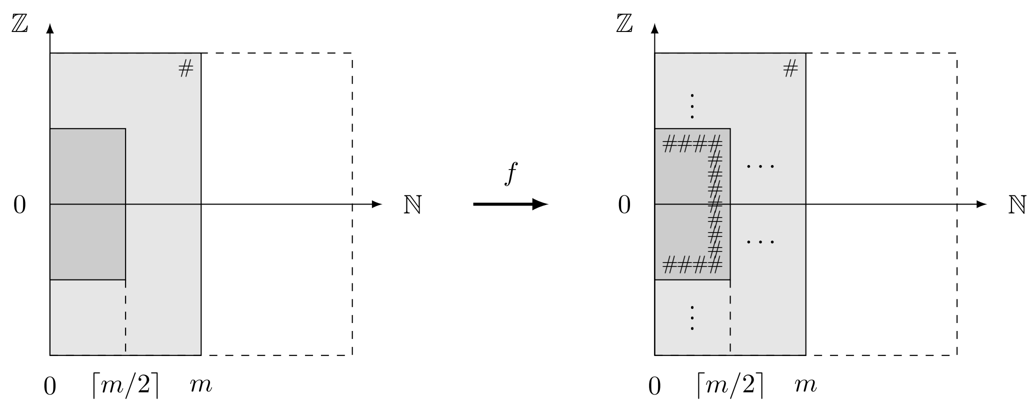


Figure 7.4: Illustration of the propagation of the error symbol $\#$ under the action of f.

3. **Computation of the entropy of f:**

   Let us compute the entropy of the function f. Since the subspace X' is stable under the action of f, and f acts like the shift map $\sigma^{\mathbf{e}^2}$ on X', the entropy of f is greater than the entropy of the dynamical system $(X', \mathbf{e}^2)$: $h(f) \geq h$.

4. **Number of patterns:**

Let us prove the reverse inequality.

For $l \geq 1$, consider the partition $\mathcal{U}_l$ of $\{0,1\}^{\mathbb{N}\times\mathbb{Z}}$ which consists of the set of cylinders on Δ_l. The minimal number of open subsets in a sub-cover of

$$\bigvee_{k \leq n} f^{-k}(\mathcal{U}_l)$$

corresponds to the number of possible patterns on for the restriction of some $f^n(x)$ on Δ_l.

For such a pattern p, there are two possibilities:

- it does not contain any symbol #; this means that the restriction of x on $\Delta_l + n\mathbf{e}^2$ is p. Since the restriction of $f^n(x)$ on Δ_l contains no symbol #, the configuration x has no symbol # on $\Delta_{2^n l}$: if this was the case, then by the propagation of the # symbol, there would be a # in the restriction of $f^n(x)$ on Δ_l. For the same reason, $x_{\Delta_{2^n l}}$ does not contain any of the $2^n l$ first forbidden patterns in the EDS.
- there is a symbol # in this pattern ; let l' be the smallest integer such that there is a # in the restriction of p on $\Delta_{l'}$. Necessarily, it appears on the border of this set. From the propagation of the error signal under the action of f, this means that this symbol has to be on the border of $\Delta_{2l'}$ in the configuration $f^{n-1}(x)$, hence the outside of $\Delta_{l'-1}$ is fully colored with # in configuration $f^n(x)$. Moreover, the restriction of p on $\Delta_{l'-1}$ is the restriction of x on $\Delta_{l'-1} + n\mathbf{e}^2$, and the restriction of x to $\Delta_{2^n(l'-1)}$ contains none of the $2^{l'-1}$ first forbidden patterns.

From this we deduce that the number

$$N_n(f, \{0,1\}^{\mathbb{N}\times\mathbb{Z}}, \mathcal{U}_l)$$

is smaller than the number of patterns on $[\![0,l]\!] \times [\![-l, l+n]\!]$ which are included in in a $\Delta_{2(l+n)}$ pattern for n great enough (since $\Delta_{2(l+n)} + n\mathbf{e}^2$ is contained in $\Delta_{2^n l}$ for n great enough) containing no forbidden patterns amongst the n first forbidden patterns for the EDS.

5. **Upper bound on the entropy:**

This implies that

$$\inf_n \frac{\log_2(N_n(f, \{0,1\}^{\mathbb{N}\times\mathbb{Z}}, \mathcal{U}_l))}{n}$$

is smaller than

$$\inf_n \frac{\log(M_{n,n',n''})}{n},$$

where $M_{n,n',n''}$ is the number of patterns on $[\![0,l]\!] \times [\![-l, l+n]\!]$ included in some $\Delta_{l+n+n'}$ pattern which does not contain any of the n'' first forbidden patterns in the EDS. This means that

$$\begin{array}{rcl} \inf_n \frac{\log_2(N_n(f,\{0,1\}^{\mathbb{N}\times\mathbb{Z}},\mathcal{U}_l))}{n} & \leq & \inf_{n',n''} \inf_n \frac{\log_2(M_{n,n',n''})}{n} = \inf_n \inf_{n',n''} \frac{\log_2(M_{n,n',n''})}{n} \\ & = & \inf_n \frac{\log_2(N_n(\sigma_{\mathbf{e}^2},\{0,1\}^{\mathbb{N}\times\mathbb{Z}},\mathcal{U}_l))}{n} \end{array}.$$

As a consequence,

$$h(f) = \sup_l \inf_n \frac{\log_2(N_n(f, \{0,1\}^{\mathbb{N}\times\mathbb{Z}}, \mathcal{U}_l))}{n} \leq \sup_l \inf_n \frac{\log_2(N_n(\sigma^{\mathbf{e}^2}, \{0,1\}^{\mathbb{N}\times\mathbb{Z}}, \mathcal{U}_l))}{n} = h.$$

This implies that $h(f) = h$.

6. **Recoding and entropy:**

   Consider then the function $g = \phi \circ f \circ \phi^{-1}$. Since f is computable, ϕ and ϕ^{-1} are computable, then g is computable. Moreover, g is conjugated to f, hence $h(f) = h(g)$.

$\square$

7.5.2 Surjectivity constraint

The function constructed in the proof of Proposition 7.35 is clearly not onto. This is not known if this characterization could be adapted to this restriction. We have a realization of the Σ_1-computable real numbers under this constraint:

Proposition 7.36. *Let $h \geq 0$ be a Σ_1-computable real number. There exists some onto computable function from $\{0,1\}^{\mathbb{N}}$ into itself which has entropy h.*

Proof. Let $h \geq 0$ be a Σ_1-computable number. There exists a computable sequence $(h_n)_n$ of rational numbers such that $h = \sup_n h_n$.

1. **Realization of the rational numbers:**

   Let $r = p/q$ be a positive rational number, and consider the function f acting as follows: $f(0^j 1^{q-j} x) = 0^{j+1} 1^{q-j-1} x$ for $0 \leq j \leq q-1$, $f(0^q x) = 1^q \sigma^p(x)$, and if w is a word of length q not in $\{0^j 1^{q-j}, 0 \leq j \leq q\}$, $f(wx) = wx$. We have $qh(f) = h(f^q) = p$.

   As a consequence, $h(f) = p/q$.

2. **Proof the the equality** $h(f^q) = p$**:**

   Let us denote $g = f^q$, $\mathbb{U}_l$ the cover which consists of the cylinders $[u]_0$ for u a length $l \geq q+1$ word. Since the cylinders corresponding to words u beginning with w such that $|w| = q$ are stable under the action of g, for all n, the minimal cardinality of a sub-cover of

   $$\bigvee_{k \leq n} g^{-k}(\mathcal{U}_l)$$

   is the sum of the minimal cardinalities of a sub-cover of the cover

   $$\bigvee_{k \leq n} g^{-k}_{[w]_0}(\mathcal{U}_{l,w})$$

   of $[w]_0$, where $\mathcal{U}_{l,w}$ is the set of cylinders corresponding to words having w as prefix, for w length q word. For $w \neq 1^q$, this number is 2^{l-q}, and for $w = 1^q$, this number is 2^{l+pn}. This means that the entropy of g verifies the following inequalities:

   $$\inf_n \frac{\log(2^{l+pn})}{n} \leq h(g) \leq \inf_n \frac{\log(2^q 2^{l+pn})}{n},$$

   which implies that $h(f^q) = p$.

3. **Stability under supremum:**

   It follows that for all k, there exists a computable onto function f_k on the Cantor set whose entropy is h_k. Define f by $f(0^k 1x) = 0^k 1 f_k(x)$ for all $k \in \mathbb{N}$. This function is computable, onto and $h(f) = \sup_n h_n = h$. Indeed, for all l, the entropy of f relatively to the cover $\mathcal{U}_l$ is the supremum of the numbers h_k, $k \leq l$, using the same arguments as in the first paragraph.

   Since the topological entropy of f is the supremum over l of the entropies of f relatively to $\mathcal{U}_l$, this entropy if $\sup_l \sup_{k \leq l} h_k = \sup_l h_l$.

$\square$

7.6 One-dimensional continuous dynamics

In this section, the space is $X = [0,1]$. In Section 7.6.1, we provide specific tools in order to study the computability of the entropy for systems on the interval. The results presented in this section are known results of computability theory on the compact interval. For a reference, see [BHW08]. In Section 7.6.2, we give a characterization of the entropies of computable dynamical systems $([0,1], f)$, and consider a constraint on the total variation of the function. In this case, we give a characterization.

7.6.1 Computability of the functions of the interval

Definition 7.37. *Let $f : [0,1] \to [0,1]$ be a continuous function. Its **modulus of continuity** is the function $\rho_f : [0,1] \to [0,1]$ defined for all $\delta \geq 0$ by*

$$\rho_f(\delta) = \max\{|f(x) - f(y)| : |x - y| \leq \delta\}.$$

Proposition 7.38. *Let $f : [0,1] \to [0,1]$ be a continuous function. Its modulus of continuity is also a continuous function.*

Proof. 1. **The function ρ_f is sub-additive:**

For all δ_1, δ_2 such that $\delta_1 + \delta_2 \leq 1$,

$$\rho_f(\delta_1 + \delta_2) \leq \rho_f(\delta_1) + \rho_f(\delta_2).$$

Indeed, let x, y be two elements of $[0,1]$ such that $|x - y| \leq \delta_1 + \delta_2$. Without loss of generality, we can assume that $x < y$. There exists $z \in [0,1]$ such that $|x - z| \leq \delta_1$ and $|y - z| \leq \delta_2$: one can take for instance $x + \delta_1$. As a consequence,

$$|f(x) - f(y)| \leq |f(x) - f(z)| + |f(y) - f(z)| \leq \rho_f(\delta_1) + \rho_f(\delta_2),$$

and

$$\rho_f(\delta_1 + \delta_2) \leq \rho_f(\delta_1) + \rho_f(\delta_2).$$

2. Since f is a continuous function, and $[0,1]$ is compact, this function is equicontinuous. This means exactly that $\rho_f(\delta) \to 0$ when δ tends towards 0.

3. **Right-continuity:**

Let $(\delta_n)_n$ be a non-increasing sequence converging towards some $\delta \in [0,1]$. By the first point, and since ρ_f is non-decreasing, we have the inequality

$$\rho_f(\delta) \leq \rho_f(\delta_n) = \rho_f(\delta + (\delta_n - \delta)) \leq \rho_f(\delta) + \rho_f(\delta_n - \delta).$$

Since $\rho_f(\delta_n - \delta)$ tends towards 0 by last point, $\rho_f(\delta_n)$ tends towards $\rho_f(\delta)$.

4. **Left-continuity:**

Let $(\delta_n)_n$ be a non-decreasing sequence converging towards some $\delta \in [0,1]$. Since ρ_f is non-decreasing, for all n,

$$\rho_f(\delta_n) \leq \rho_f(\delta).$$

There exists $x < y$ such that $|x - y| \leq \delta$ such that $|f(x) - f(y)| = \rho_f(\delta)$. Consider $x_n = x + \delta - \delta_n$. Since $|y - x_n| = \delta_n$,

$$|f(x_n) - f(y)| \leq \rho_f(\delta_n) \leq \rho_f(\delta).$$

Since f is continuous, $|f(x_n) - f(y)|$ tends towards $\rho_f(\delta)$, and thus $\rho_f(\delta_n)$ tends towards $\rho_f(\delta)$. This ends the proof.

□

Remark 48. *This result is not true in general. For instance, consider* $Z = [0,1/4]\bigcup[3/4,1]$*, with the metrics of* $[0,1]$ *induced on this set, and* $\mathcal{S}$ *is the set of rational points in* Z*. This space is a computable metric space which is also recursively compact. Then consider the function* $f : Z \to Z$*, defined by* $f(z) = 0$ *if* $z \leq 1/4$*, and* $f(z) = 1$ *if* $z \geq 3/4$*. This function is computable. However its modulus of continuity is a function on the interval* $[0,1]$ *into itself, and* $\rho_f(\delta) = 0$ *if* $\delta < 1/2$*, and* $\rho_f(\delta) = 1$ *if* $\delta \geq 1/2$*. Since this function is not continuous, it can not be computable.*

Proposition 7.39. *When* f *is computable,* ρ_f *is a computable function.*

Proof. 1. **Sufficient conditions to verify:** Let $f : [0,1] \to [0,1]$ be a computable function. Since f is continuous, ρ_f is also continuous. As a consequence for any ideal ball B_n, and any point $x \in \rho_f^{-1}(B_n)$, there exists some ideal ball B_k containing x and contained in $\rho_f^{-1}(B_n)$. As a consequence, $\rho_f^{-1}(B_n)$ is the union of the ideal balls contained in it. As a consequence, it is sufficient to prove that the inclusion

$$B_k \subset \rho_f^{-1}(B_n)$$

is semi-decidable. Since ρ_f is non-decreasing, this is equivalent to the condition that the inequality $\rho_f(r) > t$ and $\rho_f(r) < t$ are semi-decidable for $r, t \in \mathbb{Q}$.

2. **The inequality $\rho_f(r) > t$ is semi-decidable:**

   The algorithm works as follows.

   (a) It successively computes more and more precise approximations of the distances $d(s_k, s_n)$ for k, n integers.

   (b) In parallel, for each of these ordered pairs, it computes more and more precise approximations of $d(f(s_k), f(s_n))$. This is possible, using Proposition 7.18.

   (c) In parallel, it checks if $d(f(s_k), f(s_n)) > t$ and $d(s_k, s_n) < r$. If one of these tests is true, the algorithm stops.

   This algorithm stops if and only if $\rho_f(r) > t$, since it is equivalent to the existence of k, n such that $d(f(s_k), f(s_n)) > t$ and $d(s_k, s_n) < r$. Indeed, even if the maximum in the definition of $\rho_f(r)$ is reached for two elements such that the distance between them is equal to r, this value can be approximated with arbitrary precision with two ideal points whose relative distance is $< r$. This is due to the fact that X is the compact interval.

3. **The inequality $\rho_f(r) < t$ is semi-decidable:**

   The function ρ_f verifies the following equality, for all δ:

$$\rho_f(r) = \inf\{t' : F_r \subset \phi^{-1}([0,t'[)\},$$

   where $\phi : X \times X \to [0,+\infty[$ is defined by $\phi(x,y) = d(f(x), f(y))$, and

$$F_r = \{(x,y) \in X^2 : d(x,y) \leq \delta\}.$$

   This is a computable closed subset of $X \times X$, and since d and f are computable, ϕ is also computable. As a consequence of Proposition 7.13, and the definition of a computable function, the inclusion

$$F_r \subset \phi^{-1}([0,t'[)$$

   is semi-decidable, since in order to tell if this inclusion is true, this it is sufficient to find some finite sets of integers I, J, such that $\mathcal{U}_{I\times J}$ covers F_r and the balls $B_n \times B_k$, $(n,k) \in I \times J$, are all included in $\phi^{-1}([0,t'[)$.

As a consequence, $\rho_f(r) < t$ is semi-decidable, since it is sufficient to tell that this inequality is true, to find some t' such that

$$F_r \subset \phi^{-1}([0, t'[).$$

$\square$

Lemma 7.40. *Let $f[0,1] \to [0,1]$ be a computable function. There exists an algorithm, which, given as input some (n,k,l), outputs m such that*

$$|f^k(s_n) - s_m| \leq 2^{-l}.$$

Proof. For all n, k, and m integers,

$$|f^{k+2}(s_n) - f^{k+1}(s_m)| \leq \rho_f(|f^{k+1}(s_n) - f^k(s_m)|).$$

As a consequence, for all $n, k \geq 1$ and any sequence $m_1, ..., m_k$, using the last inequality,

$$\begin{array}{rcl} |f^k(s_n) - s_{m_k}| & \leq & |f^k(s_n) - f^{k-1}(s_{m_1})| + ... + |f(s_{m_{k-1}}) - s_{m_k})| \\ & \leq & \rho_f(|f^{k-1}(s_n) - f^{k-2}(s_{m_1})|) + ... + \rho_f(|f(s_{m_{k-2}}) - s_{m_{k-1}}|) + |f(s_{m_{k-1}}) - s_{m_k})| \end{array}.$$

Since f and ρ_f are computable functions, the inequality

$$\rho_f(|f^{k-1}(s_n) - f^{k-2}(s_{m_1})|) + ... + \rho_f(|f(s_{m_{k-2}}) - s_{m_{k-1}}|) + |f(s_{m_{k-1}}) - s_{m_k})| < 2^{-l}$$

is semi-decidable. Hence there is an algorithm which on input (n,k,l) outputs m such that

$$|f^k(s_n) - s_m| \leq 2^{-l},$$

by checking this successively for all the finite sequences $(m_1, ..., m_k)$ until finding one such the inequality is verified. $\square$

Lemma 7.41. *Let $f : [0,1] \to [0,1]$ be a computable function. The inclusion*

$$f(\overline{B_k}) \subset B_n$$

is semi-decidable.

Proof. Let us describe some algorithm which semi-decides this inclusion. Let (k,n) be some input for this algorithm. Here are the steps of the algorithm:

1. Using Lemma 7.40, for all the values of m, consider the finite sequence $(\tilde{s}_i)_{i=1..m}$ of rational numbers regularly spaced such that the smallest and the largest are the extreme points of B_k.
2. For each of these sequences, the algorithm computes more and more precise approximations of the numbers $f(\tilde{s}_i)$.
3. In parallel, it computes more and more precise approximations of the image of modulus of continuity of f on the value equal to the space between the numbers $\tilde{s}_i$.
4. If at some point, if all the numbers than consist of the sum of the approximation of the modulus with any of the approximation of the values $f(\tilde{s}_i)$ are in B_n, then the algorithm halts.

This algorithm halts if and only if the inclusion

$$f(\overline{B_k}) \subset B_n$$

is true, since this inclusion is true if and only if the conditions on the approximations are true when these approximations are sufficiently precise. $\square$

7.6.2 Characterization of the entropies of computable dynamical systems on the interval

We prove here the following theorem:

Theorem 7.42. *The numbers that are the entropy of a computable dynamical system* $([0,1], f)$ *are exactly the* Σ_1*-computable non-negative numbers.*

The proof of this theorem relies mostly on the notion of horseshoe presented in Definition 7.45.

In Section 7.6.2.1, we present the tools from interval dynamics used for the proof of Theorem 7.42. The proof is presented in Section 7.6.2.2.

7.6.2.1 Interval dynamics

Definition 7.43. *We say that a continuous map* $f : [0,1] \to [0,1]$ *is piecewise monotone when there exists* k *and some numbers* $x_1 = 0 < ... < x_k = 1$ *such that on each* $[x_i, x_{i+1}]$ *the restriction of* f *on this interval is monotone. When* f *is strictly monotone on each interval* $[x_i, x_{i+1}]$*,we say that* f *is strictly piecewise monotone.*

Lemma 7.44. *Let* $f : [0,1] \to [0,1]$ *be a continuous strictly piecewise monotone function. For all* n*,* f^n *is strictly piecewise monotone.*

Proof. We use a recurrence argument. Since this property is true for $n = 1$, this is sufficient to prove that if f and g are two strictly piecewise monotone functions, then $f \circ g$ is also strictly piecewise monotone. Let f, g be two such functions. There exists k and some numbers $x_1 = 0 < ... < x_k = 1$ such that on each $[x_i, x_{i+1}]$ the restriction of f on this interval is increasing or decreasing, and there exists k' and some numbers $x'_1 = 0 < ... < x'_{k'} = 1$ such that on each $[x'_i, x'_{i+1}]$ the restriction of g on this interval is strictly monotone. Let us denote $(x'')_{i \leq k''}$ the finite increasing sequence such that

$$\{x''_i\} = \{x_i : i \leq k\} \bigcup \{x'_i : i \leq k'\}.$$

On each of these intervals, both f and g are strictly monotone, and as a consequence, $f \circ g$ is also strictly monotone. □

The following notion is a powerful tool to analyze dynamical systems on the interval, and in particular the topological entropy of these systems.

Definition 7.45. *Let* f *be a continuous map of the interval* $[0,1]$*. A* (p,n) *horseshoe of* f *is a finite collection of compact disjoints intervals* $J_1, J_2, ..., J_p$ *such that for all* i*,* $f^n(J_i)$ *contains a neighborhood of* $\cup_j J_j$*. The horseshoe is called monotone when on every* J_j*,* f^n *is strictly monotone.*

The proof of the following theorem can be found in [Rue15]. This theorem will be used for the proof of the obstruction.

Theorem 7.46 ([MS77])**.** *Let* f *be a continuous map of the interval* $[0,1]$*. Then*

$$h(f) = \sup_{(p,n)\in\Delta} \frac{\log_2(p)}{n},$$

where $(p,n) \in \Delta \Leftrightarrow f$ *admits a* (p,n)*-horseshoe. When* f *is a piecewise monotone map,*

$$h(f) = \sup_{(p,n)\in\Delta'} \frac{\log_2(p)}{n},$$

where $(p,n) \in \Delta' \Leftrightarrow f$ *admits a* (p,n) *monotone horseshoe. The number* $\log_2(p)/n$ *is called the entropy the* (p,n)*-horseshoes.*

Definition 7.47. *Let $f : [0,1] \to [0,1]$ be a continuous map. Its* ***variation*** *is the following number*

$$V(f) = \sup \sum_{k=0}^{n} |f(x_k) - f(x_{k+1})|,$$

where the finite sequence $(x_k)_k$ is such that $x_0 = 0 < ... < x_{n+1} = 1$.

The following lemma is stated as Corollary 15.2.14 in [KHL95]:

Lemma 7.48 ([KHL95])**.** *Let $f : [0,1] \to [0,1]$ be a continuous map. Then we have the following equality:*

$$h(f) = \lim_n \frac{\log_2(V(f^n))}{n}$$

7.6.2.2 Proof of Theorem 7.42

Theorem 7.42 follows from the two following propositions.

Proposition 7.49. *For all $f : [0,1] \to [0,1]$, $h(f)$ is Σ_1-computable.*

Remark 49. *By convention, $+\infty$ is considered as a Σ_1-computable number.*

Idea: *the idea of this proof is to construct an algorithm which detects if a set of open intervals with rational extreme points is a (p,n)-horseshoe of f for some (p,n), and if this is the case, outputs $\frac{\log_2(p)}{n}$. Since the entropy is the supremum of these numbers, this is a Σ_1-number.*

For I and J two intervals, we denote $I < J$ when for all $x \in I$ and $y \in J$, $x < y$.

Proof. If the entropy of f is $\log_2(p)/n$ for some integers p and n, then it is a Σ_1-computable number. Let us assume in the following that it is not the case.

Given as input an integer n, consider the algorithm described as follows:

1. It initializes a counter $c = 0$, and a variable $d = 0$.

2. It enumerates all the finite sets of open intervals with rational extremal points $(I_i =]l_i, r_i[)_{1 \leq i \leq k}$ such that $I_1 < ... < I_k$, and the intervals $\overline{I_i}$ are disjoint. For each of these sets, it runs the algorithm of Lemma 7.41 to semi-decide if the inclusions

$$\overline{I_i} \subset f(I_j).$$

3. When alle these conditions are true for some set of open intervals and $c < n$, and if $d < \log_2(k)/p$ for the current k, p, then d is changed to $\log_2(k)/p$, and c is changed to $c + 1$.

4. when $c = n$, the algorithm stops and outputs d.

The output on input n is denoted d_n.

On every input, since the entropy of f is not equal to the entropy of any of its horseshoes, f admits an infinity of horseshoes with strictly increasing entropy. Since the counter is incremented each time detecting a horseshoe with rational extremal points, this algorithm terminates on every input.

Any horseshoe with rational extremal points is detected by the algorithm on input n sufficiently large. Since in the statement of Theorem 7.46, the horseshoes can be considered with rational extremal points, from this theorem from Theorem 7.46, the entropy of f is $\sup_n d_n$, which is a Σ_1-computable number. □

The following proposition gives a realization of all the Σ_1-computable numbers, completing the characterization.

Proposition 7.50. *Let $h \geq 0$ be a Σ_1-computable real number. There exists a computable function $f : [0,1] \to [0,1]$ such that the entropy of $([0,1], f)$ is h.*

Proof. 1. **Realization of computable numbers:** First, we prove that given h a positive computable real number, there exists a continuous function f_h of the interval $[0,1]$ such that $f_h(0) = 0$, $f_h(1) = 1$ such that the topological entropy of f_h is h. Since for any map f, $h(f^n) = nh(f)$, and that this is trivial to realize entropy zero, this is sufficient to prove this for $h \in]0,1]$.

Denote $s = 2^h$, which is also a computable real number, since the exponential map is computable, and consider the function f_h defined as follows:

- when $x \leq \frac{1+s}{4s}$, then
$$f_h(x) = sx,$$
- when $\frac{1+s}{4s} \leq x \leq \frac{3s-1}{4s}$, then
$$f_h(x) = \frac{1+s}{4} - s\left(x - \frac{1+s}{4s}\right),$$
- and when $x \geq \frac{3s-1}{4s}$,
$$f_h(x) = s\left(x - \frac{3s-1}{4s}\right) - \frac{3s-1}{4}.$$

See an illustration of Figure 7.5.

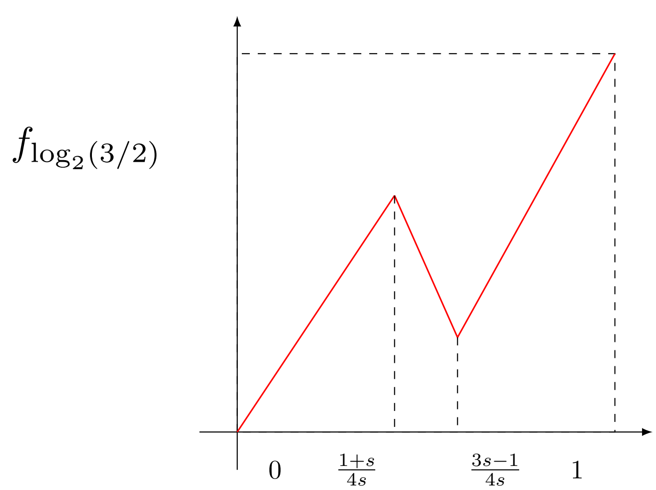


Figure 7.5: Illustration of the map f_h when $s = 3/2$.

Since on each of the intervals defining f_h, this function has slope s, for all n, the function f_h^n has slope s^n. Hence the variation of f_h^n is s^n. Indeed, from Lemma 7.44, there exists a sequence $x_1 = 0 < ... < x_k = 1$ such that on each of the intervals $[x_i, x_{i+1}]$, this function is increasing or decreasing, and the slope of the function is s^n. Considering a finite sequence $y_1 = 0 < ... < y_{k'} = 1$,

$$\sum_{j=1}^{k'-1} |f_h^n(y_{j+1}) - f_h^n(y_{j+1})| \leq \sum_{j=1}^{k''-1} |f_h^n(\tilde{y}_{j+1}) - f_h^n(\tilde{y}_{j+1})|,$$

where the sequence $\tilde{y}$ is defined to be the increasing sequence such that

$$\{\tilde{y}_i : i \leq k''\} = \{x_i : i \leq k\} \bigcup \{y_i : i \leq k'\}.$$

On each of the intervals $[\tilde{y}_{j+1}, \tilde{y}_{j+1}]$, the slope of the function is s^n, hence

$$|f_h^n(\tilde{y}_{j+1}) - f_h^n(\tilde{y}_{j+1})| = s^n|\tilde{y}_{j+1} - \tilde{y}_{j+1}| = s^n(\tilde{y}_{j+1} - \tilde{y}_{j+1}).$$

This imply that the variation of f_h^n is

$$V(f_h^n) = \sup s^n \left(\sum_j (\tilde{y}_{j+1} - \tilde{y}_{j+1}) \right) = s^n.$$

From Lemma 7.48, we deduce that the entropy of f_h is

$$h(f_h) = \lim_n \frac{\log_2(s^n)}{n} = h.$$

The function f_h is computable: indeed, since this function is continuous, it is sufficient to prove that the inclusion

$$f_h(]r, l[) \subset]r', l'[$$

is semi-decidable, where r, r', l, l' are rational numbers. Since an extremal point of $f_h(]r, l[)$ is sq, $\frac{1+s}{4} - s\left(q - \frac{1+s}{4s}\right)$, or $s\left(q - \frac{3s-1}{4s}\right) - \frac{3s-1}{4}$ for some rational number q, then it is sufficient to have that the inequalities $s > q$ and $s < q$ are semi-decidable, where q is a rational number. This is true, since s is computable.

2. **Realization of Σ_1-computable numbers:** Let now h be a positive Σ_1-computable real number, and (h_n) an increasing computable sequence of rational numbers such that $h_n \to h$.

   Let f be the map defined by, for all $x, n \geq 0$, if $x \in \left[\sum_{k=1}^{n} \frac{1}{2^k}, \sum_{k=1}^{n+1} \frac{1}{2^k}\right]$,

   $$f(x) = \sum_{k=1}^{n} \frac{1}{2^k} + \frac{1}{2^n} f_{h_n}.$$

   See an illustration on Figure 7.6.

   The variation of f^n is

   $$V(f^n) = \sum_{k=1}^{+\infty} \frac{1}{2^k} h_k^n,$$

   hence we have the following inequalities, for all p, since $(h_k)_k$ is increasing:

   $$\sum_{k=1}^{p} \frac{1}{2^k} h_k^n + \frac{1}{2^{p-1}} h_p^n \leq V(f^n) \leq \sup_k h_k^n = h^n$$

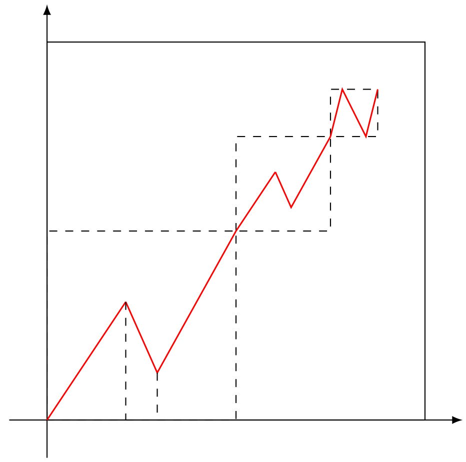


Figure 7.6: Illustration of an example of map f, when $h_1 = h_2 = \log_2(3/2)$ and $h_3 = 1$.

This implies that the entropy of f_h verifies

$$h_p \leq h(f) \leq h,$$

and $h(f) = h$.

The function f is computable. Indeed, it is sufficient to see that the inequality $f(q) < r$ and $f(q) > r$ are semi-decidable, where q, r are rational numbers.

An algorithm which allows to semi-decide the first one (it is similar for the second one) works as follows:

(a) it first checks if one of the rational numbers is 0 or 1. In this case, it is trivial to decide the inequality.

(b) if none of these rational is 0 or 1, then the algorithm computes some n such that

$$\sum_{k=1}^{n} \frac{1}{2^k} \leq q \leq \sum_{k=1}^{n+1} \frac{1}{2^k}.$$

(c) then it computes more and more precise approximations of the number h_n - this is possible since $(h_n)_n$ is uniformly computable - checks if the number

$$2^n (q - \sum_{k=1}^{n} \frac{1}{2^k})$$

is in one of the intervals defining the function f_{h_n}. If this is the case, the algorithm computes the image, and checks if this image is $< r$. If this is the case, the algorithm stops.

This algorithm stops when the inequality is verified: this comes from the fact that $(h_n)_n$ is computable.

□

Remark 50. *One could wonder if there is a similar characterization for the entropy dimension. From our knowledge, such a characterization is not known. Moreover, we lack tools to analyze this question, since the answer for the entropy relies mostly on the notion of horseshoe, and there is no equivalent tool in order to analyze entropy dimension.*

7.6.3 Computability of the entropy under constraint on the variation.

In this section, we prove that under some restriction on the variation, the entropy of a system on the interval becomes computable, and we provide realizations.

The interest in this question comes from the work of R. Pavlov and M. Shraudner in [PS15], where the authors prove that all the entropies of block gluing multidimensional SFT are computable numbers. They make then a first step towards the characterization of the real numbers that are the entropy of a block gluing $\mathbb{Z}^3$-SFT, by realizing a subclass of the class corresponding to the obstruction.

In a more general way we would like to understand which conditions on the dynamical system imply that the entropy is computable.

For all real number s, we denote $\mathcal{FS}_s$ the class of piecewise linear continuous maps $f : [0,1] \to [0,1]$ such that the slope of f is $\pm s$ on each of the intervals where it is linear.

Theorem 7.51. *The numbers that are the entropy of a computable map in $\mathcal{FS}_s$ for some real number s are the computable numbers.*

Proof.

- If f is in $\mathcal{FS}_s$ for some real number s and is computable, then s is a computable number. Indeed, if $x_1 = 0 < ...x_k = 1$ is such that on each $[x_i, x_{i+1}]$, the restriction of f is linear, then considering a rational number r such that $x_1 < r < x_2$, there is an algorithm which on input n outputs some rational numbers r_n and r'_n such that $|r_n - f(r)| \leq r.2^{-n-1}$ and $|r'_n - f(0)| \leq r.2^{-n-1}$. Hence
$$|(r_n - r'_n) - (f(r) - f(0))| \leq |r_n - f(r)| + |r'_n - f(0)| \leq r.2^{-n},$$
which means that
$$\left| \frac{r_n - r'_n}{r} - s \right| \leq 2^{-n}.$$
Since $h(f) = \log(s)$, then the entropy of f is computable.
- Reciprocally, since for any function f an any integer n, $h(f^n) = nh(f)$, this is sufficient to prove that all the numbers in $[0,1]$ are the entropy of a function which verifies the conditions of the theorem. This is true, considering the functions constructed in the first point of the proof of Proposition 7.50.

□

7.7 Comments

The restrictions on dynamical systems considered in the last section seem excessive: can we improve this result by relaxing this condition, in order to understand what are the optimal conditions so that the entropy of a computable system becomes computable ? One could consider using Theorem 7.52, due also to M. Misiurewicz, in order to compute from above the entropy under some monotonicity condition. However this is not trivial to compute the number of maximal intervals on which f^n is monotone.

This theorem can be found in [Rue15]:

Theorem 7.52 ([MS77]). *Let f be a continuous map $[0,1] \to [0,1]$ which is piecewise monotonic. Then the entropy of f is given by*
$$h(f) = \inf_n \frac{\log_2(c_n(f))}{n},$$
where $c_n(f)$ is the number of maximal intervals on which f^n is monotone.

Moreover, it would be interesting to extend the investigation of this chapter further to other computable metric spaces that we did not examined here : one-dimensional continua, manifolds, etc.

Chapter 8

Comments

Sommaire

In this chapter, we comment on the previous ones. We mostly deal here with non-formalised ideas, relating our results and constructions to other problems, of mathematical and biological natures. We also provide directions for further research.

8.1 Computational threshold effect

We studied in Chapter 4 and Chapter 5 the limit between the computable and the uncomputable through a quantification of appearance properties in subshifts. We characterized a threshold for decidable subshifts with a summability condition. As a consequence, the class of functions that are under (or above) the threshold is not given in an explicit way.

8.1.1 Explicit threshold

In order to have a more explicit way to express the limit, we could use families of functions with real parameter: for instance $f^\alpha : n \mapsto n^\alpha$, for $\alpha \geq 0$. In this case, this is not difficult to see that the limit parameter is $\alpha = 1$. Thus we understand well the case of decidable subshifts, in any dimension. However, we would like to understand this limit for subshifts of finite type. We proved that the entropy of $\mathbb{Z}^2$-SFT is computable in the sub-logarithmic regime, and can be any Π_1-computable number in the linear regime. Thus a gap remains. For this problem, we expect the following:

Conjecture 2. *Let $\alpha < 1$ be a computable number. The entropy of any f^α-block gluing $\mathbb{Z}^2$-SFT is computable.*

The intuitive argument behind this conjecture is that it is hard to embed universal computation into subshifts of finite type having dynamical restrictions without the use of hierarchical structures. These structures impose, as in the case of the Robinson subshift, a linear or super-linear gap function.

An intermediate step towards Conjecture 2 would be to know if there exists an intermediate (sharp) gap function of SFT in this gap:

Question 3. *Is there a computable function $f : \mathbb{N} \to \mathbb{N}$ such that $\log(n) = o(f(n))$ and $f(n) = o(n)$ and a subshift X which is sharp f-block gluing?*

8.1.2 Dependance of the threshold on dimension

A priori the threshold (if it exists) depends on the dimension of the subshifts.

For three-dimensional subshifts of finite type, we have even less information. This is not hard to see that that the entropy can still be any Π_1-computable in the linear regime. However, we don't know any obstruction. For instance:

Question 4. *Is the entropy of three-dimensional (constant) block gluing $\mathbb{Z}^3$-SFT computable ?*

An intermediate question is the following one, for which no answer is known yet:

Question 5. *If a $\mathbb{Z}^3$-SFT is (constant) block gluing, does it have a periodic point ? If true, is the set of periodic points dense ?*

Although we are not able to prove it, we believe that the answer to this question is negative. In our construction, we imposed the linear block gluing using a distortion operator which gives a posteriori some flexibility to the construction. We need more tools that allow to include this flexibility in the rules defining the SFT. This would allow the hierarchical structures to be imposed in the configurations of the subshift while letting a choice in the development direction of this hierarchical structures.

Moreover, it could be interesting to extend this investigation to subshifts on other finitely generated groups, in order to understand the effect of geometry of the group on the treshold.

8.2 Limits of the fixed-point paradigm

8.2.1 Block gluing with gap function

The characterization of the entropies of *transitive* multidimensional SFT was also proved in [DR17a] using the fixed-point paradigm. However, we proved a stronger result in Chapter 4: the difference with the fixed point construction are:

1. the block gluing property (the fixed point construction is only net gluing),

2. and the linearity of the gap function.

Moreover, the linear regime seems to be limiting for the use of the fixed-point paradigm. We explain here this statement.

In the construction of [DR17a], the delimitation of the computing units and their construction are achieved by the machines themselves. This means that at any scale, the machines build the other ones at scale immediately above. That is the reason why they all need to end their computations in time. For this purpose, the machines impose the size of the cells to be great enough. The size given to the cells is

$$\prod_{k=1}^{n} 3^{C^k} = \nu \lambda^{C^n},$$

where $C > 1$ is an integer which is taken large enough, $\nu = 3^{\frac{C-2}{C-1}}$ and $\lambda = 3^{\frac{C}{C-1}}$.

Remark 51. *In contrast, in the constructions using the Robinson subshift as structure layer, the machines execute their computation without interacting. They do not need to end their computations since any step of computation is reached by the machines in computing units high enough in the hierarchy. As well, in [AS13], dynamical restrictions are prevented by the cooperation between the machines, which imposes rigidity to the construction.*

The completion results can be adapted here to the completion of a m-block into a block having size

$$\nu \lambda^{C^{\left\lceil \log_3(\log_C(m)) \right\rceil + 2}}.$$

The gluing set of one m-block relative to another one thus contains positions regularly spaced with space smaller than

$$3^{C^{\left\lceil \log_3(\log_C(m)) \right\rceil + 3}} \leq m^{C^4}.$$

Moreover, the $m_n = \nu.\lambda^{C^n}$-blocks constituting the subshift are exactly spaced by

$$\nu \lambda^{C^{n+1}} \geq \nu^{1-\frac{1}{C}} m_n^C$$

As a consequence, these subshifts are net gluing with a gap function framed by two super-linear polynomial functions. As well, there are upper restrictions on the size of the cells. Thus, although the fixed-point paradigm is powerful, it works in a precise range of possible gap functions for the net gluing property.

In other words, this construction relies on the low density of the areas supporting computations. The linear regime imposes that these areas are more dense. There are thus more degenerated behaviors, and another way to simulate them is needed. This is what our construction provides. It relies on the fact that in the Robinson subshift, the size computing units depends exponentially on the level (instead of a double exponential in the fixed-point construction). One can imagine to impose more control on the growth of the computing units sot that this growth is slower. This way we could construct aperiodic net gluing subshifts with sub-linear gap function. However, there would be a limitation for sub-linear block gluing coming from the distortion operators.

8.2.2 Minimality

In Chapter 6, we characterized the possible entropy dimensions of $\mathbb{Z}^3$-SFT under the constraint of minimality as the Δ_2-computable numbers in $[0, 2]$.

The difference with the construction of Chapter 4 is the necessity to alternate all the possible local displays of random bits generating the entropy dimension.

In the construction of [DR17a] the transitivity was achieved by simulating in all the computing units of the degenerated behaviors of the machines. This was possible since their number is sufficiently small (the number of behaviors that can be simulated is limited by the size of the computing units). We don't know if the same adaptation of the fixed-point constructions is possible for sets of random bits generating an entropy dimension. As a consequence, we don't know if this result could be achieved with the fixed-point paradigm and the idea of simulation of degenerated behaviors.

Another way to achieve this result with this paradigm would be to adapt the simulation result in [DR17b] in order to simulate two dimensional effective subshifts and simulate the subshifts constructed in [JLKP17]. It would be left to prove that these subshifts are or can be adapted to be effective when the number realized as entropy dimension is Δ_2-computable. Moreover, Lemma 1 in [DR17b] allows the combination of the minimality of the computing architecture and the minimality of the simulated subshift into a global minimality. The main argument in the application of this lemma can be stated as the sub-action of a minimal one-dimensional subshift is also minimal, which is specific to one-dimensional subshifts and does not applies directly to bi-dimensional ones.

Moreover, our construction in Chapter 6 relies on the fact that Fermat numbers are co-prime. Thus the technique developed there can not be adapted directly to lower complexities (for instance when the complexity function is bounded by a polynomial), since the counters impose that the entropy dimension to be positive.

8.3 Multidimensional SFT having closed form entropy

One of the most widely studied SFT is the **hard core model**, defined as the subshift on $\mathbb{Z}^2$ and on alphabet $\{0,1\}$ defined by forbidding 1 symbols to be superimposed on two adjacent positions of $\mathbb{Z}^2$. We don't know explicitely its entropy:

Question 6. *Can we express the entropy of the hard core model with a formula ?*

There are still not a lot of other subshifts of finite type whose entropy is known to be given by a formula. In Chapter 4, we used a family of non-trivial SFT $(\Delta'_r)_{r\geq 1}$ whose entropies are given by:

$$h(\Delta'_r) = \frac{\log_2(1+r)}{r}.$$

Although we are not able to give a formula for the entropy of the hard core model, this suggests an approach to compute the entropy of particular subshifts. We used this in order to compute the entropy of the subshifts Δ'_r, $r \geq 1$: by adding random bits on structures appearing in the configurations of the subshift in order to make the entropy easy to compute, with the cost of some parasitic entropy that can be reduced to be arbitrarily small.

8.4 Comparing continuous and discrete dynamics

In Chapter 7, we have seen that there is a remarkable difference between the interval and the Cantor set with respect to the computability of the entropy of their computable dynamical systems. On one hand, the entropy of effective one-dimensional subshifts is Π_1-calculable. This comes from the fact that these systems have a generating cover. As a consequence, there exists an ideal open cover such that the topological entropy of the system is the entropy relative to this cover, which is an infimum. In other words, it suppresses the supremum part of the general formula of topological entropy. On the other hand, on the interval, the entropy of a computable dynamical system on $[0,1]$ is Σ_1-computable. In order to prove this obstruction, we relied on the notion of horseshoe, and the relation of the topological entropy to horseshoes.

However, the comparison between the Cantor set and the interval suggests that there is a complementary simplification. This means that for computable maps of the interval the infimum part of the entropy formula is suppressed. We thus address the following question:

Question 7. *Is the entropy of a computable dynamical system on the interval relative to an ideal open cover computable ?*

As the topological entropy of an interval map is related to horseshoes, it is probable that the answer to this question involves a relation of open covers to horseshoes.

8.5 Discrimination of dynamical properties using topological invariants

Although not proved here, one can deduce from the results presented in Chapter 4 the following statement: *the only number that is the entropy dimension of a linearly block gluing $\mathbb{Z}^3$-SFT, for any non-trival linear function, is $+\infty$.*

Thus the two invariants studied here (topological entropy and entropy dimension), discriminate the two properties of linear block gluing and minimality.

in the following sense:

1. The set of possible entropies of linearly block gluing $\mathbb{Z}^3$-SFTs for any non-trivial linear function is non trivial.

2. The set of possible entropy dimensions of minimal $\mathbb{Z}^3$-SFTs is non trivial.

3. The set of possible entropies of minimal $\mathbb{Z}^3$-SFTs is trivial.

4. The set of numbers that are entropy dimension of a linearly block gluing $\mathbb{Z}^3$-SFTs for any non-trivial linear function is trivial.

One can imagine to extend this into a correspondence between a broader set of growth-type invariants (as presented in Chapter 2) and appearance properties (presented in the same chapter). One could modify for instance the sequence of quantifiers defining these properties. Such a correspondence could be interpreted as a way to measure dynamical properties with topological invariants.

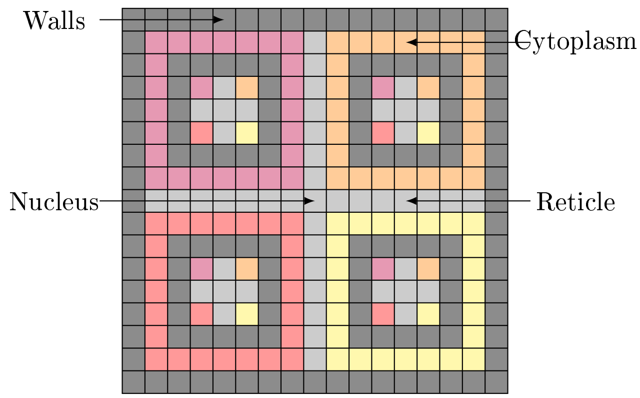
8.6 Analogies

In this section we give explanations (Section 8.6.1) about the vocabulary used in this text, in Chapter 4 and Chapter 6. This comes from the elementary description of living cells. In Section 8.6.3, we explain in what sense the phenomena observed in the constructions, listed in Section 8.6.2, are natural. In Section 8.6.4 and Section 8.6.5, we provide ideas relative to the question of the function of non-coding parts of the DNA, and the trade-off between wire and diffusive communication modes, through these analogies.

8.6.1 On the analogy with living cells

We used, along this text, a coherent vocabulary inspired from elementary sub-structures of (eukariotic) living cells. These words make these sub-structures correspond to sub-structures of the computing units used in most of the constructions embedding computations in subshifts of finite type. This correspondence relies on the function of these sub-structures in the whole construction.

This correspondence is on the following figure:



Here is an explanation of this:

- The function of walls in our construction is to draw the limits of the cells, as the walls of a living cell, and in particular the areas where the computing machines evolve.
- The cytoplasm is the area where the machines work. In the living cells, this area is where the proteins are produced from the RNA. In our construction, the RNA corresponds to the machine head. The other symbols might be amino acids, which by diffusion and reaction with the RNA, act on the basis symbols. These ones can thought of as the resulting proteins.
- The nucleus is the center of the eukariotic cell, and contains some information which codes for the construction of RNA. This in turn allow proteins to be built. In our construction, we called the center position of the computing units nucleus. The reason of this analogy is that this position contains some information (that we called DNA) that codes for the behavior of the computing machines. The DNA acts on the machines through direct implication but also indirect implication, by retroaction of the error signals. This might be a limit of the analogy: a priori, no retroaction of this type of an organism on its own DNA is known.

Following this analogy, we observe that the $\texttt{on} - \texttt{off}$ and $\texttt{in} - \texttt{out}$ signals act on the implication of symbols on the reticle that code for the behavior of the machine (the ADN acts on them and they act themselves on the machines). This is similar to the phenomenon of inhibition of the genes expression.

Let us note that the notion of coding in these construction refers to the logical implication of the symbols placed on some position from other symbols and local rules. As a consequence, the coded behavior is not necessarily posterior in time to the coding information. Moreover, in these systems, there is no mechanism writing on the DNA by logical implication. We could although imagine some systems where this happens. What kind of constraints could imply this phenomenon ?

These structures appeared natural in order to adapt to dynamical constraints, approaching the threshold from above. We expect that approaching the threshold further would enforce more and more complex structures of this kind.

Moreover, we expect that an attentive examination of the robustness of these constructions and their relation to dynamical constraints could provide an intuition of the strategical interest of these structures.

An objection to these analogies could be that the model does not have any time dimension. One answer to this issue would be that we usually think the logical implication process of the local rules as forming this dimension of time.

8.6.2 Phenomena induced by adaptation to constraints

According to the constructions, we could observe various other phenomena exhibiting strong similarities with biological ones:

- centralization in the computing units of the information coding for the behavior of the machines (Chapter 4, Section 4.5.4) ; there is a strong analogy with the role of DNA in cells.

- synchronization mechanisms inside the computing units, and not outside where communication is inhibited (Chapter 4),

- specification of propagation direction of error signals (Chapter 6, Section 6.4.7.3),

- the use of two types of transmission of the information: wire transmission and diffusive one. An example of diffusive communication can be found for instance in Chapter 6, Section 6.4.5.2, with the freezing signal; an example of wire communication can be found in Section 6.4.7.3 with the error signals. Moreover, the two modes of communications are complementary in the computing machine model used in this construction.

- In some counters that code for the behaviors of the machine, the presence of non-coding parts (as in DNA) which serve for the global structure of the system (Chapter 6,Section 6.4.5.1).

- The functional specialization of computation units: these units are divided into sub-units that are attributed with a specific function (Chapter 4 and Chapter 6).

- Modification of the computing machines mechanism:

  1. the computation areas are composed of lines and columns transmitting information and on whose intersections the computation steps of the machines are executed. This was already used in R. Robinson's construction [Rob71], and then in M. Hochman and T. Meyerovitch's construction [HM10] ;
  2. some lines and columns are active, and some others not (Section 6.4.6.1); following the analogy with DNA, this would be similar as the phenomenon of regulation of gene expression.
  3. the model includes a shadow state, meaning the absence of any machine head (Section 6.4.6.2), in order to have a precise quantity of possible initializations of the machines,
  4. the computation steps occur at intersections of active columns and lines, and they are described as a set of outputs going to specified directions, given as a function of inputs also coming from specified directions (Section 6.4.6.3).

Could we understand these phenomena in a more systematic way ?

We expect that a variation on the invariants and dynamical constraints considered in the text would lead to the observation of various other phenomena that could enrich this point of view.

8.6.3 Naturalness of the observed phenomena

8.6.3.1 Centralization of the coding information

In Chapter 4, the centralization of the coding information seems natural in the sense that the possible variations of the principle used in this construction involves more complex behaviors.

Indeed, one could imagine not dividing the computing units and use a communication networks such that the space between computing units and and this network is large except on the leftmost bottommost position of the computing units. On these positions, the network touches them. This network allows, together with a signal that propagates through it, the behaviors of the machines in the computing units to alternate.

We don't follow this idea in this text since this is more complex to be implemented with local rules. Moreover, the gap function for the net gluing property would be greater because of the space needed between the network and the cells.

8.6.3.2 Hierarchical structures

In the construction of Chapter 4 and Chapter 6 we rely on hierarchical structures, like all the other constructions embedding autonomous computation in subshifts of finite type, or constructing strongly aperiodic subshifts of finite type in abstract groups. In this case, the possibility of building hierarchical structures often derives from the properties of the group. In particular, the construction of [DR17a] uses hierarchical structures.

The aperiodicity is usually thought as the first ingredient for embedding computations in SFT. However, from the previous observation, it is natural to address the following question:

Question 8. *Is this possible to embed (autonomous) computation in SFT under constraints without using hierarchical structures ? In particular, is it possible to embed computations in an aperiodic SFT which does not exhibit hierarchical structures, such as Kari-Culik subshift [Cul96] ?*

A hint that the answer to these questions is negative is that these hierarchical structures are deeply involved in the constructions. They allow the organization of some global behavior by logical implications through an infinite tree structure. On some nodes of this structure the communication is inhibited, allowing some freedom that ensures the dynamical restrictions. Let us note that an answer to this question could provide us with an insight on the wide presence of hierarchical structures in the living.

8.6.4 On the coding and non-coding

In Chapter 6, the linear counters code for the behavior of the machines. The change from the construction of Chapter 4 is that the quantity of coding information is increased. Indeed, in the second construction it includes all the information determining the machines' trajectories in the cytoplasm. We need to communicate this information to the adjacent cells in order to alternate between all the possibilities. That is why the linear counters are used. Moreover, in order to ensure the minimality property, we needed that the periods of the counters in different scales are coprime. For this purpose, we complete the information with some non-coding information so that the the periodicities of the counters form a subsequence of the sequence of Fermat numbers. The added information is involved not in local properties (coding) but in the global properties of the system (minimality). The analogy with the DNA suggests that the non-coding parts of the DNA - introns - are involved in global properties of the organism - the kind of mechanisms by which it is possible would remain to be understood.

8.6.5 Trade-off between wire and diffusive communication modes

We used in our constructions the two types of communication.

Wire communication was used for all the hierarchical processes of the construction of Chapter 6, for the various signals that propagate into the structures and for communication between cells. We used this mode when we needed to transmit bits of information so that only coherent ones coincide, ensuring dynamical properties.

Diffusive communication was used mostly for synchronization processes, as in the construction of Chapter 4. There we had to synchronize the frequency bits in the quarters of the cells. It was also used and in the construction of Chapter 6 for the freezing signal. We needed it when the degenerated versions of the structures that support the information that we needed to synchronize were not connected, breaking the synchronization.

It is noteworthy that these two modes are used in a complementary way in the constructions, as well as in living systems, in synaptic connections for the wire communication, and the immune system for the diffusive one.

8.7 Measuring the organization of dynamical systems

Our work in Chapter 4 and Chapter 6 suggests another approach to the problem raised by K. Peterson and B. Wilson [PW15], which consists in finding a way to measure the **organization** property of dynamical systems, in order to account for the complexity of living systems. We first present the problem and the approach of [PW15], and then suggest other approaches.

8.7.1 On the organization of systems

Organized systems are the result of an equilibrium between coherence and freedom, stability and adaptability to new contexts.

Intuitively, a set of similar living organisms unites into an organized system in order to execute the function of preserving these living organisms. This organization involves the restriction of freedom of individual organisms through their inter-dependency, but also the increase of freedom of the whole.

In subshift of finite type constructions, we observe that there is co-dependence between two positions when the symbol on the first position restricts the possibilities or determines the symbol on the second one. These two positions are independent when all the ordered pairs of symbols on these positions are admissible. In the above sense, organization could be interpreted as an equilibrium between dependence and independence of pairs of positions.

In this context, full shifts are not organized (since all the pairs of positions are independent) as well as subshifts that consist of two constant configurations: in this case, all the positions are dependent. As a consequence, the topological entropy is not adapted to account for this kind of complexity. An adapted invariant would have high values for organized systems and low values for non-organized ones. This is not true for the entropy, since full shifts have high entropy. This is noteworthy here to recall that the concept of entropy comes from thermodynamics formalism.

8.7.2 Neural complexity and intricacy

The neuroscientists Tononi, Sporns, and Edelman [EST94] proposed a quantity, called **neural complexity**, as a way to account for the equilibrium between local segregation of neurons (into small functional groups, and thus independence among these groups), and global integration (introducing dependence).

The definition of this quantity relies on the entropy as a quantity of information. It is expressed as a sum of relative entropies of bipartitions of the system.

A similar quantity was defined by Petersen and Wilson [PW15] in the topological dynamics formalism, called **Intricacy**. This definition is, as for the entropy, based on the intricacy related to an open cover. The supremum of these intricacies is the topological entropy. However, the intricacies relative to an open cover seem of interest. The formula giving the intricacy of a subshift X relative to the open cover $\mathcal{U}_0$ which consists of the set of cylinders given by the letters of the alphabet is the following:

$$I(X,\mathcal{U}_0) = \frac{1}{2}\sum_{k\geq 1} \frac{\log(N_k(X))}{2^k} - h(X).$$

Although the value of this quantity is low for extremes subshifts - full shifts and constant ones- this quantity seems more to measure the convergence speed of the complexity function than the organization of the subshift. We suggest to study the minimal number of finite patterns that allow a given subshift of finite type to be defined. This quantity is low for the extremes subshifts and high for subshifts that are intuitively complex, like most of the SFT constructions in the literature.

8.7.3 Measuring dynamical properties with invariants

Another approach would be to develop the correspondence between dynamical properties and topological invariants that we discussed above.

The idea is that a correspondence like this could provide an insight on the way to construct an invariant that measures a property which balances freedom and structure.

This could also provide an insight on how dynamical properties of subsystems are measured by a living system, and, following A. Damasio, on the mechanisms of emotional processes.

We also observe that in these constructions, in particular in the construction of Chapter 4, the dynamical constraints correspond to an increase of freedom, in the sense that every finite space-time diagram is a priori possible in the finite cells. However, the error signals maintain the initial restriction on these diagrams, with a little increase in the complexity of the mechanism that allows this control. A comparative study of the balance between freedom and control in these constructions and the construction in the literature could provide an insight in the way that we could associate a quantity to the property of organization.

8.7.4 On the meaning of topological invariants

In order to understand how topological invariants of a dynamical system measure its dynamical properties, it would be interesting to give more meaning to these invariants that would relate them directly to information transmission.

On one hand, the mathematical definition of the entropy appeared in the work of C. Shannon [Sha48], in which the meaning of this quantity is clear: it counts the asymptotic quantity of bits that a source can emit per time unit. This means that the entropy measures the **capacity** that the source has to transmit information to a receiver. Moreover, the fact that this quantity is the only one that verifies natural properties make it natural to define.

Through the generalisation of this quantity to topological dynamical systems, the meaning of this invariant was changed. In the context of symbolic dynamics, it sill measures a capacity of the system. However, this is a capacity *to produce complex trajectories*: this is reflected by the fact that the entropy of a system is the supremum over the Kolmogorov complexities of the trajectories of its elements [Sim15]. This interpretation of the entropy as a capacity is also coherent with the fact that it is equal to the Hausdorff dimension of the subshift [Sim15].

On the other hand, other topological invariants, such as the entropy dimension [Mey11] [DHKP11] or polynomial growth rate [Mey11] were defined by similarity with the entropy. The difference with the entropy is that these invariants focus on systems having lower complexity function. However, to our knowledge, there is no interpretation of these invariants similar to those of the entropy relating it to topology (Hausdorff dimension) or information theory (Kolmogorov complexity). Do they have a meaning in terms of information transmission as C. Shannon's entropy?

List of Figures

Bibliography

[AB12] N. Aubrun and M.-P. Béal. Tree-shifts of finite type. *Theoretical Computer Science*, 459:16–25, 2012.

[ACCR14] C. Alcocer-Cuaron, V. Castano, and A.L Rivera. Hierarchical structure of biological systems. *Bioengineered*, 5:73–79, 2014.

[AFSM95] L. Agnati, K. Fuxe, I. Strömberg, and Zoli M. Intercellular communication in the brain: wiring versus volume transmission. *Neuroscience*, 69:711–726, 1995.

[AK13] N. Aubrun and J. Kari. Tiling problems on baumslag-solitar groups. *Electronic proceedings in Theoretical computer science*, 128, 2013.

[AKM65] R. L. Adler, A. G. Konheim, and M. H. McAndrew. Topological entropy. *Transactions of the AMS*, 114:309–319, 1965.

[AS13] N. Aubrun and M. Sablik. Simulation of effective subshifts by two-dimensional subshifts of finite type. *Acta Appl. Math.*, 126:35–63, 2013.

[AS14] N. Aubrun and M. Sablik. Multidimensional effective s-adic subshift are sofic. *Uniform Distribution Theory*, 9, 2014.

[AW70] R.L Adler and B. Weiss. Similarity of automorphisms of the torus. *Memoirs of the American Mathematical Society*, 98, 1970.

[Bar17] S. Barbieri. A geometric simulation theorem on direct products of finitely generated groups. Dynamical systems, 2017.

[BC08] O. Bournez and M.L. Campagnolo. *A Survey on Continuous Time Computations*, pages 383–423. Springer New York, New York, NY, 2008.

[Ber66] R. Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66, 1966.

[BGP13] O. Bournez, D.S. Graça, and A. Pouly. *Turing Machines can be efficiently simulated by the General Purpose Analog Computer*, pages 169–180. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[BHW08] V. Brattka, P. Hertling, and K. Weihrauch. *A Tutorial on Computable Analysis.*, pages 425–491. Springer, New York, NY, 2008.

[BJ09] E. Ben-Jacob. Learning from bacteria about natural information processing. *Annals of the New York Academy of Sciences*, 1178:78–90, 2009.

[Bow78] R. Bowen. *On Axiom A diffeomorphisms*. Published for the Conference Board of the Mathematical Sciences by the American Mathematical Society Providence, 1978.

[BPS10] M. Boyle, R. Pavlov, and M. Shraudner. Multidimensional sofic shifts without separation and their factors. *Trans. Am. Math. Soc.*, 362:4617–4653, 2010.

[BS16] S. Barbieri and M. Sablik. A generalization of the simulation theorem for semidirect products. Dynamical systems, 2016.

[BW17] L. Brown Westrick. Seas of squares with sizes from a pi10 set. *Israel Journal of Mathematics*, 222:431–462, 2017.

[COTA17] J. Cassaigne, N. Ollinger, and R. Torres-Avilés. A small minimal aperiodic reversible turing machine. *Journal of computer and system sciences*, 84:288–301, 2017.

[Cul96] K. Culik. An aperiodic set of 13 wang tiles. *Discrete Mathematics*, 160:245–251, 1996.

[Dam17] A. Damasio. *The stranger order of things : life, feeling, and the making of cultures*. Pantheon, 2017.

[dC97] M. de Carvalho. Entropy dimension of dynamical systems. *Portugaliae Mathematica*, 54:19–40, 1997.

[Dev03] R.L. Devaney. *An Introduction to Chaotic Dynamical Systems*. Westview Press, 2003.

[DHKP11] D. Dou, W. Huang, and K. Koh Park. Entropy dimension of topological dynamical systems. *Transactions of the American Mathematical Society*, 363:659–680, 2011.

[DR17a] B. Durand and A. Romashchenko. The expressiveness of quasiperiodic and minimal shifts of finite type. Discrete Mathematics, 2017.

[DR17b] B. Durand and A. Romashchenko. On the expressive power of quasiperiodic sft. Discrete Mathematics, 2017.

[DRS12] B. Durand, A. Romashchenko, and A. Shen. Fixed-point tile sets and their applications. *Journal of Computer and System Sciences*, 78:731–764, 2012.

[Dub98] J.-C. Dubacq. How to simulate turing machines by invertible one-dimensional cellular automata. *International journal of foundations of computer science*, 6:395–402, 1998.

[EST94] G. Edelman, O. Sporns, and G. Tononi. A measure for brain complexity : relating functional segregation and integration in the nervous system. *Proc. Natl. Acad. Sci.*, 91:5033–5037, 1994.

[Fis75] R. Fischer. Sofic systems and graphs. *Monatshefte für Mathematik*, 80:179–186, 1975.

[Fod00] J. Fodor. *The mind doesn't work that way*. MIT press, 2000.

[Gác01] P. Gács. Reliable cellular automata with self-organization. *Journal of Statistical Physics*, 103:45–267, 2001.

[GH16] S. Gangloff and B. Hellouin. Computability of the entropy of mixing subshifts. Dynamical systems, 2016.

[Gri73] C. Grillenberger. Construction of strictly ergodic systems i. given entropy. *Z. Wahrscheinlichkeitstheorie*, 33:215–217, 1973.

[Gri80] R.I. Grigorchuk. On burnside's problem on periodic groups (russian). *Functional Analysis and its applications*, 14:41–43, 1980.

[GS05] C. Goodman-Strauss. A strongly aperiodic set of tiles in the hyperbolic plane. *Inventiones mathematicae*, 159:119–132, 2005.

[GS17a] S. Gangloff and M. Sablik. Block gluing intensity of bidimensional sft: computability of the entropy and periodic points. Dynamical systems, 2017.

[GS17b] S. Gangloff and M. Sablik. A characterization of the entropy dimensions of minimal z3-sfts. Dynamical systems, 2017.

[Had98] J. Hadamard. Les surfaces à courbures opposées et leurs lignes géodésiques. *Journal de mathématiques pures et appliquées*, 4:27–74, 1898.

[Hed69] G.A. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Mathematical systems theory*, 3:320–375, 1969.

[HM38] G.A. Hedlund and M. Morse. Symbolic dynamics. *American Journal of Mathematics*, 60:815–866, 1938.

[HM10] M. Hochman and T. Meyerovitch. A characterization of the entropies of multidimensional shifts of finite type. *Annals of Mathematics*, 171:2011–2038, 2010.

[Hoc09] M. Hochman. On the dynamics and recursive properties of multidimensional symbolic systems. *Inventiones Mathematicae*, 176:131–167, 2009.

[HS08] P. Hertling and C. Spandl. Shifts with decidable language and non-computable entropy. *Discrete Mathematics and Theoretical Computer Science*, 10:73–95, 2008.

[HU06] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

[HV17] M. Hochman and P. Vanier. Turing spectra of minimal subshifts. In *Computer Science - Theory and Applications*, pages 154–161, 2017.

[JLKP17] U. Jung, J. Lee, and K. Koh Park. Topological entropy dimension and directional entropy dimension for z2-subshifts. *Entropy*, 2017.

[JV11] E. Jeandel and P. Vanier. π_1^0 sets and tilings. In *Theory and Applications of Models of Computation*, pages 230–239, 2011.

[JV13] E. Jeandel and P. Vanier. Turing degrees of multidimensional sft. *Theoretical Computer Science*, 505:81–92, 2013.

[JV15] E. Jeandel and P. Vanier. Characterizations of periods of multidimensional shifts. *Ergodic Theory and Dynamical Systems*, 35:431–460, 2015.

[Kan10] N. Kanwisher. Functional specificity in the human brain : A window into the functional architecture of the mind. *Proceedings of the National Academy of Sciences*, 107:11163–11170, 2010.

[KHL95] A. Katok, B. Hasselblatt, and Mendoza L. *Introduction to the modern theory of dynamical systems*. Cambridge university press, Cambridge, 1995.

[KM99] P. Koiran and C. Moore. Closed-form analytic maps in one and two dimensions can simulate turing machines. *Theoretical computer science*, 210:217–223, 1999.

[Kur97] P. Kurka. On topological dynamics of turing machines. *Theoretical computer science*, 174:203–216, 1997.

[LM95] D.A. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, New York, NY, USA, 1995.

[Mar85] B. Marcus. Sofic systems and encoding data. *Transactions on Information theory*, 1985.

[Mey11] T. Meyerovitch. Growth-type invariants for zd subshifts of finite type and arithmetical classes of real numbers. *Inventiones mathematicae*, 184:567–589, 2011.

[Mor21] H.M. Morse. Recurrent geodesics on a surface of negative curvature. *Transactions of the AMS*, 22, 1921.

[Moz89] S. Mozes. Tilings, substitution systems and dynamical systems generated by them. *Journal d'Analyse Mathématique*, 53:139–186, 1989.

[MS77] M. Misiurewicz and W. Szlenk. Entropy of piecewise monotone mappings. *Studia Mathematica*, 67:45–63, 1977.

[Mye74] D. Myers. Nonrecursive tilings of the plane. ii. *The Journal of Symbolic Logic*, 39:286–294, 1974.

[Pav12] R. Pavlov. Approximating the hard square entropy constant with probabilistic methods. *Annals of probability*, 40:2362–2399, 2012.

[PS15] R. Pavlov and M. Schraudner. Entropies realizable by block gluing shifts of finite type. *Journal d'Analyse Mathématique*, 126:113–174, 2015.

[PW15] K. Petersen and B. Wilson. Dynamical intricacy and average sample complexity. *Dynamical Systems*, pages 1–50, 2015.

[Rob71] R. Robinson. Undecidability and nonperiodicity for tilings of the plane. *Inventiones Mathematicae*, 12:177–209, 1971.

[Rog87] H. Rogers. *Theory of recursive functions and effective computability*. MIT Press, 1987.

[RTV91] E. Rosh, E. Thompson, and F.J. Varela. *The embodied mind : cognitive science and human experience*. MIT Press, Cambridge, 1991.

[Rud90] D.J. Rudolph. x2, x3 invariant measures and entropy. *Ergodic theory and dynamical systems*, 10:395–406, 1990.

[Rue15] S. Ruette. Chaos for continuous interval maps - a survey of relationship between the various kinds of chaos. Dynamical systems, 2015.

[Ser13] T. Serre. *Hierarchical Models of the Visual System*, pages 1–12. Springer New York, New York, NY,USA, 2013.

[Sha48] C.E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 1948.

[Sim06] J.G. Simonsen. On the computability of the topological entropy of subshifts. *Discrete mathematics and Theoretical computer science*, 8:83–95, 2006.

[Sim14] S.G. Simpson. Medvedev degrees of two-dimensional subshifts of finite type. *Ergodic Theory and Dynamical Systems*, 34:679–688, 2014.

[Sim15] S.G. Simpson. Symbolic dynamics: entropy = dimension = complexity. *Theory of computing systems*, 56:527–543, 2015.

[Spa07] C. Spandl. Computing the topological entropy of shifts. *Mathematical Logic Quarterly*, 53:493–510, 2007.

[ST13] V. Salo and I. Törmä. Constructions with countable sft. *Fundamenta Informaticae*, 126:1–38, 2013.

[Sta13] B. Stanley. Bounded density shifts. *Ergodic Theory and Dynamical Systems*, 33:1891–1928, 2013.

[Wat07] J.D. Watson. *Avoid Boring People (Lessons from a life in science).* Knopf, 2007.

[Wei13] K. Weihrauch. *Computable Analysis: An Introduction.* Springer Publishing Company, Incorporated, 2013.

[Wil73] R.F. Williams. Classification of subshifts of finite type. *Annals of Mathematics*, 98:120–153, 1973.