THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *21/10/2022* **par :**

Solène June ESNAY

Limitation de la complexité de certains invariants des sous-décalages par contraintes dynamiques et structurelles

Limitation of the complexity of some invariants of subshifts by dynamical and structural constraints

---

JURY

Valérie BERTHÉ	Directrice de Recherche, Université Paris Cité	Rapportrice
Emmanuel JEANDEL	Professeur, Université de Lorraine	Rapporteur
Ronnie PAVLOV	Professeur, University of Denver	Rapporteur
Arnaud CHÉRITAT	Directeur de Recherche, Université Toulouse III	Examinateur
Vincent DELECROIX	Chargé de Recherche, Université de Bordeaux	Examinateur
Guillaume THEYSSIER	Chargé de Recherche, Aix-Marseille Université	Examinateur
Nathalie AUBRUN	Chargée de Recherche, Université Paris-Saclay	Co-directrice
Mathieu SABLIK	Professeur, Université Toulouse III	Directeur de thèse

---

École doctorale et spécialité :
MITT : Domaine Mathématiques : Mathématiques fondamentales
Unité de recherche :
Institut de Mathématiques de Toulouse (UMR 5219)
Thèse dirigée par :
Mathieu SABLIK et *Nathalie AUBRUN*
Rapportrice et rapporteurs :
Valérie BERTHÉ, *Emmanuel JEANDEL* et *Ronnie PAVLOV*

Remerciements / Acknowledgments

Trois ans.

C'est une durée qui paraît tout à fait invraisemblable, trois ans sur un sujet, trois ans à vivre une thèse ; et puis cela fait partie du quotidien, bon gré, mal gré ; et puis ça finit, alors qu'on n'y croit plus, par se terminer.

Toute personne achevant sa thèse vous le dira : écrire des remerciements est la partie la plus difficile, intime et délicate du manuscrit. J'ai réfléchi aux miens tout au long de ces trois années – j'aurais aimé en faire quelque chose de poétique, de mémorable, de frappant – mais sans doute qu'avec ces attentes je ne les aurais jamais écrits. Personne ne nous apprend vraiment à remercier les gens à la juste valeur de ce qu'iels ont apporté à notre vie, après tout. Qu'à cela ne tienne : je vais faire de mon mieux. Cela voudra dire de nombreux paragraphes commençant par "merci à", faute d'une grande variété de tournures de phrases disponibles pour exprimer ma gratitude, mais que voulez-vous ? C'est ainsi.

Oh, and some sentences in these acknowledgments will be in English, too, so that English readers may not drown in French gibberish. Part of my brain has been wired in English in the past few years, anyway.

Mes premiers remerciements vont, sans surprise, à Mathieu et Nathalie, sans qui ce manuscrit n'aurait tout simplement jamais existé. Merci à vous deux d'avoir tenu bon avec moi, d'avoir fait de votre mieux pour vous adapter aux soubresauts de cette odyssée. Merci Mathieu de m'avoir poussée (et parfois débloquée) sur tant d'idées et de sujets qui ont rendu ma thèse plus riche, même si arriver au bout de toutes ces pistes de travail a parfois été compliqué – j'aurais aimé que ce soit moins douloureux. Merci Nathalie d'avoir fait attention à moi en dépit de la distance, d'avoir essayé de me comprendre et de me laisser le temps dont j'avais besoin, de m'avoir plusieurs fois redonné espoir lorsque rien ne semblait avancer. Merci aussi pour m'avoir lancée sur le vaste sujet des pavages sur les groupes il y a près de cinq ans, qui demeure je crois ce qui me fascine le plus de tout ce manuscrit.

Je rajoute à ce qui précède un merci tout particulier à Benjamin pour son attention et sa présence certes ponctuelle mais bienveillante - j'espère que nous aurons l'occasion de poursuivre nos travaux ensemble.

Merci aux membres de mon jury, qui ont permis que cette soutenance se fasse ; merci pour votre intérêt envers ce que j'ai pu produire sur ces trois années. Specific thanks to Emmanuel, Ronnie and Valérie who have read this entire manuscript and provided a lot of helpful feedback – several definitions and theorems are now correct thanks to them.

Merci à Vincent Delecroix pour ces quelques jours à Bordeaux dont je garde un excellent souvenir ;

merci à Guillaume Theyssier et Pierre Guillon pour plusieurs de mes passages enthousiastes à Marseille ; merci à Martin Delacourt pour cette joyeuse discussion au CIRM. Plus généralement, merci aux personnes de mon domaine pour leur intérêt et leur gentillesse : je me sens chanceuse d'appartenir à cette communauté.

Thank you to my co-authors: Mathieu and Nathalie again of course, but also Ilkka, Alonso and Etienne. Ilkka, thank you for your multitude of ideas and your deep kindness in explaining them, and in giving us time to understand them. I hope – funnily enough – that we will be able to meet in real life one of these days. Alonso, thank you for having been in the same cellular automata hellboat as me for quite a while, and for the discussions that helped in keeping us sane in a world gone awry. I wish you more stable times to come.

Etienne, merci d'être un ami avec qui je puisse à la fois co-écrire et tant rire. Faire des maths avec toi est une vraie joie, un ping-pong d'idées infiniment satisfaisant, et j'ai hâte de m'y remettre. Tu as transformé bon nombre de conférences et séminaires en fou rires ; merci aussi pour ça. Merci enfin, évidemment, pour l'odeur du café noir et le rapport d'autopsie.

Un grand merci également à mon autre adelphe de thèse, Stella, pour nos quelques conférences ensemble : merci pour le calme loin des gens et les longues discussions de niche sur canapé (et autres activités peu neurotypiques) – c'est aussi quelque chose que j'espère retrouver à l'occasion.

Mes remerciements s'étendent à moult autres gens qui ont peuplé et participé à ensoleiller mes conférences. Je veux autant que possible éviter les listes de noms par la suite, mais merci tout de même à Hubert pour être ce visage familier et sympathique qui est là à chaque fois ; thank you Jana for being someone I'm always so happy to see again ; merci à Guilhem pour sa confiance et sa gentillesse immense ; merci à Amélia pour le sentiment de sororité ; merci à plein d'autres que je recroise ponctuellement. Un merci spécifique à Antonin pour avoir été cette présence queer dans une conférence de fin du monde.

J'aimerais aussi remercier les doctorant·e·s de l'IMT pour avoir participé à créer un environnement où il fait bon vivre (ou du moins, autant que faire se peut quand tout le monde est en thèse). Je n'ai été sur Toulouse que par petits morceaux, mais retrouver ce milieu-là m'a souvent fait du bien. Merci à vous, que je vous aie rencontré·e·s dès mon stage de pré-thèse, ou durant les pique-niques printaniers de ma troisième année. Merci notamment à Michelle et Laetitia pour quelques interactions chouettes en dépit des divisions entre bâtiments.

Merci évidemment aux gens du 104 et alentours, qui ont beaucoup peuplé mon quotidien toulousain de jeux, de discussions, et de projets d'impro avortés ; que ce soit celleux parti·e·s depuis un moment comme JM, Dom et Léo (membre honoraire pour son excellent travail sur l'équation de Monge-Ampère sur les variétés kähleriennes), ou plus prochainement sur le départ. William, bravo d'avoir tenu ces trois années aussi – beaucoup de courage pour la dernière ligne droite. Maxence, merci pour ta gentillesse et tout particulièrement pour cette nuit à m'héberger. Florian, je manque de mots pour décrire tout l'enthousiasme de nos discussions (vidéoludiques à 90%, disons-le) de ces trois années – et puis merci pour Smash, le rakugo et les pizzas, sans compter l'hébergement régulier de cette dernière année. Elena, tu mérites d'être mentionnée ici aussi évidemment ; mais tu retrouveras ton nom plus bas.

Avant de passer aux personnes qui ne sont pas rattachées à ma sphère professionnelle, je me permets un petit intermède. Ce sont mes remerciements, j'ai le droit.

Je voudrais essentiellement y dire ceci :

J'ai survécu.

J'aurais mis ces trois mots en accroche de mes remerciements, si j'avais osé. Je m'étais promis de les mettre quoi qu'il en soit. Ce n'est pas courant, mais voilà : merci aux professionnel·le·s qui m'ont suivie et aidée durant ces années dont j'aurais pu, à plusieurs reprises, ne pas voir le bout.

Et,

je voulais me remercier, aussi.

D'avoir tenu bon face à tous ces instants accumulés.

Il est étrange de regarder en arrière, de penser à septembre 2019, et de réaliser comme je ne suis plus tout à fait la même personne qu'en débutant cette thèse. Certain·e·s me feront remarquer que c'est assez évident, puisque je n'étais out qu'à une poignée de gens à ce moment-là ; mais tout ceci va au-delà. Je ressens tellement d'émotions et j'aime tant de gens aujourd'hui ! Il me paraît invraisemblable d'avoir vécu si longtemps sans tout ça.

J'ai beaucoup souffert, aussi. Vivre est devenu plus intense, plus réel ; et par voie de fait plus dangereux. J'en ressors heureuse, pourtant : avec la sensation d'achever quelque chose, mais surtout d'avoir une meilleure connaissance de moi-même, de m'accepter et m'apprécier davantage, et d'être un peu plus proche de qui j'ai envie d'être.

Antoine, Florette, il me semble cohérent de commencer cette partie des remerciements avec vous. Merci, de tout coeur, de m'avoir soutenue et défendue pendant un bon nombre d'années (bigre, on est tout de même sur du plus de quinze et de dix ans, ça n'a proprement aucun sens). Merci de votre soutien durant ces trois ans écoulés, aussi, même sous forme de memes et de discussions essentiellement à distance.

Les STMN, je réalise que vous entrez aussi dans la catégorie plus de dix ans ! C'est fascinant. Merci à vous d'être toujours aussi formidables et drôles ; merci à celleux d'entre vous qui ont pu m'héberger sur Toulouse au cours de ma thèse ; merci pour toutes les joies de nos retrouvailles qui ensoleillent mes vacances, et d'encourager mes travaux de carreleuse sur gaufres.

Merci à toute la famille Bonnefous pour les années de gentillesse et de jeux, et l'accueil récent – et tout particulièrement à Laure pour les discussions à cette occasion.

Le présent paragraphe est dédié d'abord à tous les gens avec qui j'ai pu discuter via de longs mails ou messages de nouvelles au cours de ces années : merci beaucoup. Plus que tout, je me dois d'y mentionner quelques connaissances et ami·e·s de longue date qui survivent à de grands échanges de pavés malgré les longs silences radios au milieu : Gemna, Margot et Hugo. Gemna, j'espère pouvoir te revoir dans quelques mois ou dans cinq ans – c'est en tous cas toujours une joie immense, et je t'envoie énormément d'affection. Margot, j'ai hâte de partager ton enthousiasme pour des jeux que je ne parviens pas encore à faire tourner ; je t'en écrirai un jour des nouvelles dignes de ce nom. Hugo, on en reparle dans un prochain paragraphe, héhé.

Mes remerciements suivants vont à Gaspard, Quentin et Géraldine pour les séjours parisiens enthousiastes et les perles de copies. Merci aussi aux deux derniers (et à Alistair, dans une certaine mesure) pour les hébergements, les repas et les longues discussions – j'en suis très touchée.

Le paragraphe qui s'annonce concerne l'ENS et les personnes qui y sont adjacentes, et je vais être honnête : vous êtes beaucoup trop nombreuxses que j'hésite à mentionner, et je ne m'en sortirai pas si je commence, alors je ferai sans pour la majeure partie. Merci donc à toutes les personnes avec qui j'ai pu passer des bons moments au fil de ces trois ans : toi qui cherches explicitement ton nom ici, tu es sans doute concerné·e. Merci notamment énormément aux membres de la Geekorale, des Improfesseurs et de la BDthêk, et à celleux qui continuent à les faire tourner maintenant. Merci à celleux avec qui j'ai pu renouer des liens distendus, et à celleux que je n'ai pas arrêté de croiser ponctuellement en répétant qu'il faudrait qu'on se voie davantage. Merci à toutes ces personnes que je suis si enthousiaste de retrouver dans des castings. Merci aux gens des dernières InterLudes qui m'ont vue briller.

Merci à KT et Léo pour de sombres histoires de voitures mal garées. Merci à Emmy pour le coming in. Merci à Julie pour les lettres, même si j'ai failli à ma tâche ces derniers temps. Merci à Bastien et Youssef : malgré le covid, je suis vraiment heureuse d'être dans cette timeline avec vous. Merci à Dana pour la chaussette et le confort, et à Elodie pour les câlins et les cookies. Merci à Florian d'être un perpétuel générateur de joie.

Dobby, je ne savais pas trop où te mettre, alors tu es là. Merci d'être cool et drôle, et pour Chamaël.

Aux Improbilisé·e·s, que ce soient ceux présents depuis les vengeurs du sud ou les plus récemment arrivés (bébé ou non), merci d'avoir rendu les confinements plus supportables et d'être un point fixe de mes étés (mais pas merci pour les poules). Certain·e·s me sont plus proches que d'autres, mais vous êtes toustes des personnes vraiment intéressantes, et j'aime beaucoup discuter et jouer avec vous : j'espère que l'occasion se présentera encore longtemps.

Finally, thank you to all the people who cared and that I left unread one way or another – most notably, though not only, Sandy, Cristina and LolV. To those who tried to get news that way, I'm sorry Discord is such a difficult medium for me. I think of you still.

S'il me reste des personnes à remercier plus spécifiquement pour les années écoulées, merci d'abord à celles dont la présence se dessine particulièrement ces mois-ci tandis que ma thèse s'achève.

Aaren, merci d'être une boule de gentillesse et de peluches, et de toutes nos discussions qui ont pu précéder cette colocation. Nattes, merci d'avoir prouvé qu'on pouvait greffer des pinces à n'importe quoi, et d'avoir très pertinemment suggéré d'écrire cette thèse en format Excel. Je suis contente de cette année qui s'annonce avec vous.

Par ailleurs, de façon inattendue mais profondément heureuse, merci à Lisa pour la douceur et les violettes.

Pour ces trois ans, merci d'abord à Leïla et Clément, pour avoir formé un endroit safe où j'ai pu déployer mes ailes. Vous m'avez apporté beaucoup de recul et d'outils, et une partie de ma sensation d'avoir progressé vient de votre aide. Merci, merci, merci.

Elena – on commence à attaquer les remerciements difficiles à écrire, dis-donc, je peine à trouver mes mots. Merci d'avoir été une formidable coloc par intérim ; merci pour les manifs ; merci d'avoir été si longtemps mon ancre à l'IMT ; merci pour les discussions cartoonesques ; merci de me supporter.

Hugo, je t'avais annoncé un paragraphe dédié, le voici. Je pourrais te remercier pour les robots, pour Bill, pour la R&D émotionnelle jusqu'à pas d'heure, pour les mails et les appels à chapitres, et pour mille autres choses encore comme autant de témoins d'une connivence qui me sidère à chaque fois, mais (prétérition subtile) je me contenterai de te dire merci pour cette relation pleine d'humour et de complicité. Tu es la personne la plus mentionnée de tous ces remerciements, et en un sens ça ne me paraît pas absurde.

Iris, Aëto, je n'ai pas les mots pour tout ce que vous avez apporté à ce printemps. Le sentiment de communauté que vous m'avez fait ressentir m'a tant aidée ; merci de faire partie de ma vie depuis. Plus spécifiquement, merci à Iris de m'être fait me rendre compte que je suis un mochi (je ne sais plus pourquoi j'ai écrit que je devais te remercier pour ça, d'autant plus que c'est quelque chose que je savais déjà, mais voilà). And Aëto, I– I. Thank you so much for being my companion.

Before ending these lengthy thanks with the two people who purely and simply made these three years, allow me to insert a few words – I lied: a lot of words – about some people (and things) who may never read these acknowledgments yet participated, unbeknownst to them, in shaping my mental landscape and myself during the last years. So thank you, a bit haphazardly, to foxes, to blue things, to my HRT, to Pride month, to all those queer people who do covers of other queer people on their bedroom floor, to Little Light and inkskinned for their heart-wrenching poetry, to one blue koala, to Mira and BK, to our dear friend Jonathan and Dracula Daily, and to Jorts the cat for the sunbeams to come. Thank you to Supergiant Games for being the studio closest to my heart. Thank you to ZUN for all of Gensokyo, and for having provided most of my background music of the past three years. Thank you to Alice Oseman for the giddy enthusiasm, to Loika for Aucis and Shir, to Ari for the softness of her worlds and characters, to Lee and Nate for having been 2020's most precious couple, and to the latter for helping me feeling so seen – and for giving power to the word *stay*. Thank you to Rebecca Sugar for all the songs and the tears.

At last, thank you to Marceline Abadeer, Sulla Pinsky, Mae Borowski, Sorawo Kamikoshi, Maribel Hearn, and to all the worst girls since Eve.

Pour finir, merci du fond du coeur à Lauren et Lison. Vous savez ce que je vous dois.

Lauren, tu as rendu ces trois ans supportables au quotidien et les as remplis de bons moments : je n'aurais pas tenu sans toi.

Lison – que dire ? – merci pour tout le reste. Merci d'être là.

Je vous aime.

A toutes les personnes qui se découvrent fille, mathématicienne, ou les deux.

A mes adelphes qui se battent pour être.

Abstract

Etant donnés un ensemble fini de symboles et une liste de règles spécifiant lesquels d'entre eux peuvent apparaître côte à côte, on peut construire un ensemble – possiblement vide – de lignes infinies de symboles dans les deux directions, obéissant à ces règles, appelées configurations. Un ensemble de configurations est appelé un *sous-décalage unidimensionnel*, et il s'agit de l'objet mathématique au cœur de la *dynamique symbolique*. La notion de sous-décalage peut être généralisée en indexant les symboles par $\mathbb{Z}^2$ – ce qui revient à paver le plan infini discret – ou par n'importe quel groupe de type fini. De nombreuses questions peuvent être posées sur les sous-décalages, notamment s'il existe un algorithme capable de déterminer lesquels d'entre eux sont vides à partir de leurs règles ; et pour un sous-décalage non vide, s'il contient beaucoup de configurations, ou certaines particulièrement complexes. Ces questions correspondent à des *invariants de conjugaison* : le Problème du Domino, l'entropie, l'apériodicité, la complexité arithmétique du langage.

Cette thèse, subdivisée en trois parties essentiellement indépendantes, étudie comment tous ces invariants sont affectés sous différentes conditions, et comment certaines contraintes sur les sous-décalages peuvent causer des changements dans leur comportement. Dans la première partie, nous nous intéressons aux attracteurs topologiques des automates cellulaires, qui sont des sous-décalages, et montrons quelle complexité maximale ils peuvent atteindre dans la hiérarchie arithmétique. Dans la deuxième partie, nous fixons des restrictions horizontales sur les sous-décalages bidimensionnels, et souhaitons savoir si le Problème du Domino reste indécidable et quelles sont les entropies possibles pour leurs sous-systèmes de type fini. Dans la troisième partie, nous étendons la définition de sous-décalage aux groupes de type fini, et présentons trois méthodes de constructions distinctes sur les groupes de Baumslag-Solitar, montrant que leur notion d'apériodicité est plus fine que celle qui existe en deux dimensions.

Given a finite set of symbols and a list of rules specifying which of them can appear next to each other, one can build a – possibly empty – set of infinite lines of symbols in two directions obeying these rules, named configurations. A set of configurations is called a *one-dimensional subshift*, and it is the mathematical object at the core of *symbolic dynamics*. The notion of subshift can be generalized by indexing the symbols by $\mathbb{Z}^2$ – that is, tiling the infinite discrete plane – or by any finitely generated group. A number of questions can be asked about subshifts, notably if there exists an algorithm able to tell which of them are empty given their rules; and for a nonempty subshift, if it contains a lot of configurations, or particularly complex ones. All of these questions give rise to *conjugacy invariants*: the Domino Problem, the entropy, the aperiodicity, the arithmetical complexity of the language.

This thesis, divided into three mostly independent parts, studies how all these invariants are affected by different settings, and how some constraints on subshifts can cause changes in their behavior. In the first part, we focus on topological attractors of cellular automata, which are subshifts, and find their maximal complexity in the arithmetical hierarchy. In the second part, we fix horizontal restrictions on two-dimensional subshifts, and try to know whether the Domino Problem stays undecidable and the possible entropies for their subsystems of finite type. In the third part, we extend the definition of subshift on finitely generated groups and present three distinct construction techniques on Baumslag-Solitar groups that show how their notion of aperiodicity is sharper than the one in two dimensions.

Introduction

A Beginner's Guide to Dominoes

Consider a box of dominoes, in which there are infinitely many copies of each domino. Suppose you want to create an infinite straight line of dominoes, respecting their usual adjacency rules. There are a lot of such lines – infinitely many, actually. Now, suppose you are given a box with two additional difficulties. First, dominoes have an orientation now, marked one way or another, that you have to respect; so dominoes 4 | 2 and 2 | 4 are considered entirely separate types. Second, your box misses some types of dominoes – not copies, but entire types, for instance dominoes labeled 4 | 2 are nowhere to be found. A lot of fascinating questions arise from this simple thought experiment:

1. Does a given box allow even a single infinite line of dominoes that respects the adjacency rules, or are you stuck, as you fail to extend any pattern you come up with in one direction or the other?

2. Can you find a systematic method so that whatever box is given to you, a glance at the available types of dominoes is enough for you to tell if a valid line exists?

3. Can you always build periodic lines? That is, if a valid line exists for a given box, can another, "simpler" line be built which simply repeats a pattern over and over?

4. Somewhat conversely, can you build lines that are, in some sense, very complicated to describe?

5. "How many" different lines of dominoes are available for a given box? Of course, this "number" will most likely be infinitely many, but can we somewhat quantify that infinity – by looking at how many segments of finite size n exist for a growing n?

These questions, as it turns out, are already mathematical. To get a better grasp of them, and bring some answers, consider a small step in formalization, that changes little in appearance: merely replace each type of domino by a specific symbol, and the adjacency rules of the dominoes by a literal list of "which symbol can be put next to which one". Congratulations: you have built a *subshift*[1], the set of all the biinfinite[2] lines that are written with your symbols and obey your adjacency rules – these lines are formally called *configurations*. Now, name $\mathcal{A}$ your set of symbols – also called the *alphabet* – and $\mathcal{F}$ the list of patterns you forbid, and finally, $X_{\mathcal{F}}$ the resulting subshift. With this,

[1]More precisely, it is a nearest-neighbor Subshift of Finite Type (SFT), because it uses a finite list of patterns you forbid, and restricts only adjacent symbols. As seen in the next paragraph, the general framework can use longer successions of symbols, and an infinite list of forbidden patterns.

[2]Infinite in both directions.

you just formalized the space of all the potential lines a given box of dominoes can build, and have started your dive into *symbolic dynamics*.

If you consider your construction with care, you may realize that the use of symbols allows for more combinations than domino boxes did – though both end up being equivalent. First, you can build an apparently wider variety of subshifts: consider the symbols 0 and 1, and 11 as the only forbidden adjacency; no replacement of 0 by some domino and 1 by some other can replicate all the available configurations in this. Indeed, 01, 10 and 00 are all available adjacencies, but 11 is not, which is not a behavior grasped by a mere one-to-one domino replacement: you need to change your alphabet to three dominoes (0|0, 0|1 and 1|0) to obtain the same subshift. Second, why restrict yourself to adjacent symbols in your rules for building configurations? You can look further and forbid two symbols at a given distance! As it is, $\mathcal{F}$ can also forbid the pattern where you ban the apparition of three 1's in a row, for instance. Here is a slightly more complex example, before moving forward: the subshift with alphabet $\{0, 1\}$ and $\mathcal{F} = \{10^n1 \mid n \in \mathbb{N}_0\}$. Any distance between two 1's is forbidden, and so the only configurations of $X_{\mathcal{F}}$ are the one with only 0's, and all the ones with a lonely 1 at a given position among 0's everywhere else.

Now that the formalism has been introduced, all the previous questions can be asked using it, and answers given.

1. For given $\mathcal{A}$ and $\mathcal{F}$, is $X_{\mathcal{F}}$ empty? This question, that requires a case-by-case answer *per se*, is the motivation for the next, systematic one.

2. Is there an algorithm able to tell you for any $X_{\mathcal{F}}$ if it is empty, when taking $\mathcal{A}$ and $\mathcal{F}$ as inputs? This is the *Domino Problem*, a question for which the answer here is *Yes* (Theorem 5). Its history is detailed in Part II of this thesis, along with its formal definition (Definition 4.8).

3. If $X_{\mathcal{F}}$ is nonempty, does it always contain a periodic configuration? Once again, the answer here is *Yes* . Generalizations of it are the focus of the Part III of the present thesis, using a wider formalism (Definition 7.6).

4. The complexity of the available configurations of a given $X_{\mathcal{F}}$ requires extended answers, captured by two distinct notions. The first is pattern complexity, which counts how many patterns of a given size exist. It is a rich concept, mostly unused here; see for instance [MH38, Cas00, CN10, KM19] for reference. The second is *arithmetical complexity* (Section 1.3), which tells to which extent it is impossible for an algorithm to describe anything – notably the patterns in a subshift. The answer in general simply depends on the patterns you forbid *ex nihilo*; in Part I of the present thesis, we study to which extent we can *build* complicated subshifts as attractors of a well-known kind of dynamical system, *cellular automata* (Definition 1.1).

5. Though naively-described earlier, the exact concept of how the pattern complexity of a subshift $X_{\mathcal{F}}$ grows for bigger and bigger patterns is precisely captured by the *entropy* (Section 4.2.2), a real number embodying how many choices are available when building configurations of $X_{\mathcal{F}}$. In the present setting, all entropies are somewhat "nice" numbers (Theorem 7); a short history of the notion is given in the introduction of Part II.

Is that it? Not at all! We can strive for wider contexts, and ask ourselves if our questions still make sense in these settings, and if their answers change. For example, what if we used two-dimensional dominoes – simply put, squares with four colored sides, also known as *Wang tiles* – and tried to fill the infinite plane with them, just as before? Well, we would have invented *two-dimensional subshifts*. With a little leap of imagination, and a bit more formalism (as the human brain is notoriously bad with dimensions four and higher[3]), we can actually define d-dimensional subshifts for any $d \in \mathbb{N}$. Subshifts can even be built on groups (as detailed in Part III, of which it is the focus)!

This is, overall, the motto of the present thesis: if we add constraints on subshifts, how are the answers to the previous questions modified? In what follows, using various approaches, we provide some progress on this overarching question.

Elements of Context

Though we keep most of the context specific to each of the following parts in its dedicated introduction, we cannot talk about symbolic dynamics without any mention of Morse and Hedlund's seminal papers and correspondence in the 1940s that gave birth to this branch of mathematics (see notably [MH38], and [CN06] for a detailed historical summary). The main idea behind their articles – to encode trajectories of dynamical systems on surfaces as a sequence of symbols that represents their orbit – gave rise to subshift spaces in one dimension, and to various types of mixing and structural properties on them such as topological transitivity, topological mixing and ergodicity. These have spawned numerous articles, and an entire field of study.

Similarly, this introduction cannot go on without crediting Wang, who catalyzed the study of two-dimensional subshifts through Wang tiles [Wan61] – originally meant for logical formulas – and raised fundamental questions on the Domino Problem and aperiodicity in dimension 2. Those were answered by his student Berger [Ber66], but led to various other constructions, notably by Robinson [Rob71] and Kari [Kar96], which remain the main available tools for any question related to the Domino Problem today.

The difference in behavior between one-dimensional and two-dimensional subshifts has been widely documented: the Domino Problem is decidable in one dimension but not in two [Rob71,Kar07]; there are *subshifts of finite type* (SFT, subshift with a finite set $\mathcal{F}$ of forbidden patterns) with no periodic configuration in dimension 2; entropies in one dimension are computable, but not all of them are in two dimensions [HM10]. Some interesting properties also appear in higher dimensions – see for instance the differentiation between weak and strong aperiodicity starting with dimension 3, mentioned in Section 7.2.

All these elements spark the same broad question: why exactly do complex behaviors – aperiodicity, undecidability, high pattern or arithmetical complexity – appear in some circumstances and not in others? Where is the boundary, and how can we explain these obvious qualitative differences in several invariants depending on the setting? Of course, a broad answer would be that the richer the dynamical system, the richer its possibilities. That is true; but can we tie this more precisely to actual elements of a system – the dynamical properties, the underlying structure?

To what extent does a given constraint put on a subshift restrict its complexity?

Structure of this Thesis

This thesis is divided into three parts, each of them dedicated to a different approach in answering our central question. Each of the parts begins with a detailed introduction to the subject, along

[3] Except if your name is Alicia Boole Stott.

with a chapter of defining its central notions (Chapters 1, 4 and 7). The object(s) of study of each part is broadly given by the following table.

Property \ Constraint	**Generic limit set of cellular automaton ($\mathbb{Z}$)**	**Projective restriction ($\mathbb{Z}^2$)**	**Underlying group**
Domino Problem	Always decidable	Section 5.1.2 and Section 6.2.2	Not studied but evoked in Section 7.2.2
Aperiodic SFT	Does not exist	Section 6.2.1	Chapter 8
Realizable entropies	(not studied)	Section 5.2 and Section 6.3	(not studied)
Arithmetical complexity	Chapter 2 and Chapter 3	(not studied)	(not studied)

Part I is dedicated to building arithmetically complex one-dimensional subshifts, as attractors of (one-dimensional) cellular automata. Various notions of attractors coexist in the literature of dynamical systems [Mil85]; some of them are understood well enough so that they have induced entire classifications [CIPY89, Hur90a, Hur90b, Hur92], others are still under investigation. Here we study the generic limit set, a topological attractor that is always a subshift in the context of cellular automata, and we try to grasp how complex it can be. As local transformations are repeatedly applied to configurations, which ones of these remain, or are approached asymptotically? What patterns do the transformations end up forbidding? Can we build arbitrarily complicated subshifts as attractors? What if we add dynamical constraints on the cellular automaton?

By adapting constructions previously applied to other kinds of attractors, we obtain a fair amount of new results on arithmetical complexity, some of the most important being the following:

Theorem (Theorem 3)**.** *There exists a cellular automaton f such that its generic limit set $\widetilde{\omega}(f)$ has a Σ_1^0-complete language, and $f|_{\widetilde{\omega}(f)} = \mathrm{id}|_{\widetilde{\omega}(f)}$.*

Theorem (Corollary 3.16)**.** *There exists a cellular automaton f such that $\widetilde{\omega}(f)$ has a Π_2^0-complete language, with $f|_{\widetilde{\omega}(f)} = \sigma|_{\widetilde{\omega}(f)}$ the left shift map, and $\widetilde{\omega}(f)$ containing no topological subattractor.*

Theorem (Corollary 3.21)**.** *A one-dimensional minimal subshift is the generic limit set of a cellular automaton if and only if it is chain mixing and has a Δ_2^0 language.*

Part II develops mostly new approaches to understanding the gap in behavior between one-dimensional and two-dimensional subshifts. Its main idea is to consider a one-dimensional subshift and to look at all the ways it can be extended to two dimensions; said otherwise, it studies all two-dimensional subshifts whose horizontal constraints obey some pre-set rules. This idea is notably explored in [JKM07] by studying the one-dimensional subshift obtained from a base two-dimensional subshift when projecting horizontally: it is shown that the projected subshift has entropy higher than the original. Furthermore, an article by Desai [Des06] partly characterizes the accessible entropies for a two-dimensional subshift when restrictions are added, which has indirect ties with our question.

Through Chapter 5 we investigate the matter in details, and determine more precisely the accessible entropies, along with how the Domino Problem is impacted when restricting ourselves to specific one-dimensional constraints – an adapted Domino Problem called DP_h.

Another direction we explore in Chapter 6 is how two one-dimensional subshifts can be assembled – one as horizontal rules, one as vertical rules – to form a complex resulting two-dimensional subshift. This notion has been introduced in [LMP13] under the name *axial product*, though with a focus on a specific kind of entropy. Here we use it to precise, in some sense, the frontier between 1D and 2D: we uncover how building complexity, in the forms of an undecidable Domino Problem (called DP_I) and noncomputable entropies, can be deeply shaped by "mere" one-dimensional constraints.

Our most relevant results of that part are the following:

Theorem (Theorem 9)**.** *Let H be a one-dimensional SFT. Then $DP_h(H)$ is decidable if and only if H contains only eventually periodic configurations.*

Theorem (Theorem 11)**.** *Let H be a one-dimensional SFT. The accessible entropies for two-dimensional SFTs obeying H as horizontal rules are exactly the set $[0, h(H)] \cap \Pi_1^0$, where Π_1^0 is the set of right-recursively enumerable real numbers.*

Theorem (Theorem 13)**.** *Let H be a nearest-neighbor one-dimensional SFT.*

$$DP_I(H) \text{ is decidable} \Leftrightarrow \widetilde{\mathcal{G}}(H) \text{ verifies condition } D$$

with condition D defined in Definition 6.1.

Theorem (weak version of Proposition 6.12)**.** *There exists a one-dimensional nearest-neighbor SFT $\tilde{H}$ with $DP_I(\tilde{H})$ decidable, and another one-dimensional SFT $\tilde{V}$ obtainable algorithmically from the first one, so that the entropy of the combined subshifts into a two-dimensional SFT $X_{\tilde{H},\tilde{V}}$ is not computable.*

Part III contributes to a growing trend: building subshifts on groups, and seeing how unexpected their behavior can be. This started with adaptations of two-dimensional tilings to the discrete hyperbolic plane [Kar94, GS10, Mar08] then to Baumslag-Solitar groups, whose structure is close to the hyperbolic plane [AK13], before spreading to finitely generated groups in general [CP15, Jea15, CGS17, Coh17, ABJ18, ABM19, BMP22]. While similar topics have been studied from the point of view of ergodic theory, and the aperiodicity we investigate ties into free actions on Cantor sets and Bernoulli shifts, subshift theory differentiates two kinds of aperiodicity (weak and strong), and brings the Domino Problem into consideration. These notions allow us to classify groups when building subshifts on their structure. Though the aperiodicity and the undecidability of the Domino Problem may seem tied at first glance – both arise in two dimensions, and none in one – there are groups where they are not, or at least not as much as expected [Pia08].

The third part of the present thesis contributes to charting which kinds of aperiodicities are obtainable on Baumslag-Solitar (BS) groups, refining the earliest construction on the subject and paving the way for other recent returns to these formative groups [AK21, ABHT22]. Using various approaches, we obtain the three following results:

Theorem (Theorem 17)**.** *For every $n \geq 2$, the Baumslag-Solitar group $BS(1, n)$ admits a strongly aperiodic SFT.*

Theorem (Theorem 18)**.** *For every $n \geq 2$, the Baumslag-Solitar group $BS(1, n)$ admits a weakly-but-not-strongly aperiodic SFT.*

Theorem (Theorem 20)**.** *For every $n \geq 2$, $BS(n, n)$ admits a strongly aperiodic SFT.*

Contents

II Subshifts with Projective Restrictions 53

III Subshifts on Finitely Generated Groups 113

Part I

Subshifts as Generic Limit Sets of Cellular Automata

Consider a finite set of symbols $\mathcal{A}$, and all of the biinfinite – infinite in both directions – words that can be written with them: these are called configurations, and are elements of the set $\mathcal{A}^{\mathbb{Z}}$. A simple manner of defining maps on $\mathcal{A}^{\mathbb{Z}}$ is through local rules that transform each symbol of a configuration into another symbol, depending on the state of its neighbors up to some fixed distance. Transforming all symbols at the same time by following these rules maps any configuration of $\mathcal{A}^{\mathbb{Z}}$ on another one, resulting in a map f from $\mathcal{A}^{\mathbb{Z}}$ to itself. Endowed with a topology, $(\mathcal{A}^{\mathbb{Z}}, f)$ becomes a well-known kind of discrete dynamical system: a cellular automaton.

Mostly known in popular culture through Conway's Game of Life, which is a specific two-dimensional cellular automaton, *cellular automata* (CA) were introduced in the 1940s by Ulam and Von Neumann. A widely studied class of dynamical systems, they are both simply describable and behaviorally rich. One-dimensional cellular automata, the core of this thesis' first part, have deep ties with combinatorics on words and subshift theory, that are close topics from discrete mathematics, but also with theoretical computing models, including Turing Machines – and these ties will be of use in the upcoming part.

The notion of "attractor" emerged from the wider field of dynamical systems in the 60s, as an attempt to capture and describe some asymptotic properties of a given system. As its early history is summarized by Milnor in a paper from 1985 [Mil85], he points out how various notions of attractor have been coexisting in the literature. The most-studied of these notions is the *omega-limit set*, made of all configurations that appear infinitely often or are approached with any accuracy in the system as a whole. Milnor defines the *likely limit set* and the *generic limit set* as the smallest closed sets that attract most configurations – considered individually, this time – up to a subset of starting points that is negligible, either in the measure-theoretical or topological sense.

Though these notions will be detailed in Chapter 1 of the present thesis, let us give an intuition of them to the reader. We focus on one of the simplest cellular automata as our dynamical system here: the automaton with alphabet $\{0, 1\}$, and a local rule that turns any cell to 1 at time $k + 1$ if at least one of its adjacent cells is 1 at time k, and leaves it as is in any other case. Consequently, the following local patch in any biinfinite word would evolve as follows under the global rule f:

$$\dots 100011 \dots \xrightarrow{f} \dots 110111 \dots \xrightarrow{f} \dots 111111 \dots .$$

Put simply, the 1's "spread" to adjacent cells; but not all notions of attractor will capture this behavior the same way.

In the omega-limit set, configurations $\dots 111 \dots$ and $\dots 000 \dots$ are included, but so are all configurations of the form $\dots 0011 \dots$ or $\dots 1100 \dots$ with a divide between all 0's on one side and all 1's on the other. The first two appear in it with little surprise, as they are stable configurations through the iterations of f. For the remaining ones, notice that any starting configuration with only finitely many 1's to the left will locally look like some $\dots 0011 \dots$ at some point – and so, at the scale of the system, better and better approximations of these configurations are visited when f is iterated. As a consequence, the aforementioned configurations $\dots 0011 \dots$ and $\dots 1100 \dots$ are considered attractive, even though none of them is approached by any single starting point.

The generic limit set and the likely limit set retain any configuration x so that there exists a nonnegligible set of starting configurations with each starting configuration approaching x individually (along with the closure of this set of x's). Therefore, configurations of the form $\dots 0011 \dots$ and $\dots 1100 \dots$ are not included. Furthermore, the generic limit set also bans configurations approached by a topologically negligible ("meager") set of points: as such, $\dots 000 \dots$, as the only configuration approaching itself, is not in the generic limit set. In the end, the latter solely consists of the full-1's configuration $\dots 111 \dots$.

The likely limit set is similar, but depends on a probability measure μ to establish which sets of points are negligible. As such, one could use the measure that gives weight only to $\dots 000 \dots$ to conclude that the corresponding likely limit set is merely $\{\dots 000 \dots\}$. However a Bernoulli measure, in this particular example, would yield the same attractor as the generic limit set.

A fourth, measure-based attractor, called the μ-limit set [KM00], consists of all configurations that contain only the words that keep a positive probability of appearing overall as time increases. As such, words that appear at any step but with increasing rarity will not be kept in that attractor.

Omega-limit sets have been intensively studied on cellular automata: a classification of all automata based on their omega-limit set was established in 1989 by Culik, Pachl and Yu [CIPY89]. Numerous additional articles precising the attainable complexity of the omega-limit set of a CA have been written since then [Kar92, Kar94, Maa95, BGK11]. Similarly, various notions of attractors have been studied with considerations for ergodic theory: Hurley originally proposed a classification of all CA based on a very weak notion of attractor [Hur90a, Hur90b, Hur92], which was later generalized by Kůrka [Kůr97]. The μ-limit set is a relatively recent example of that measure-theoretic focus, and various results on its complexity have been obtained during the past decade [BDS10, DPST11, BDP$^+$15, DMS18].

These results on μ-limit sets have opened new perspectives on generic limit sets, by adapting the tools from proofs on the former to constructions of the latter. The first part of the present thesis falls within this exact scope. Several results have been obtained recently [Tör20, Del21, Tör21] that begin to quantify "how complex" a generic limit set can be, depending on the initial constraints on the cellular automaton. These results mirror similar ones obtained on omega-limit sets and μ-limit sets (as highlighted in Section 2.1 and particularly Table 2.1). All of them use the same vocabulary, called the arithmetical hierarchy (see Section 1.3.2), that embodies how difficult it is to compute a given set based on the alternation of quantifiers used to describe it.

A notable element in studying generic limit sets is how they can all be described given a list of forbidden patterns that appear in none of their configurations [DG19] – for instance, in our aforementioned example, we could describe the generic limit set simply by forbidding the one-letter word 0. That property is general, though some generic limit sets may need an extremely complex list of forbidden patterns instead. As mentioned in the introduction, these sets defined by a list of forbidden patterns are *subshifts*.

The Chapter 1 of this present thesis is dedicated to any reader who would not know all the necessary notions about cellular automata, subshifts and arithmetical complexity. In the other two chapters of the present part, we prove new bounds on the complexity of the language of generic limit sets under some structural constraints. Most of the related proofs are twofold: first, find a suitable logical formula to describe the set, with as few quantifiers as possible (this is the focus of Chapter 2); second, prove that some well-built cellular automaton really has a generic limit set with the exepected description, so that the logical formula describing it cannot be simplified further in the case studied (this is the focus of Chapter 3).

We show that the language of a generic limit set on which the function f merely translates the configurations – acting as a shift on them – is Π_2^0. We also prove that the generic limit set's language is Σ_1^0 if the cellular automaton has equicontinuity points. Finally, we show that both bounds are tight. As an aside, from our most technical construction of Section 3.3, we also obtain several results about which subshifts can be realized as generic limit sets (see Section 2.3 and Section 3.4), that we think are interesting in their own right.

Chapter 1

Cellular Automata and One-Dimensional Subshifts

This chapter is a formal summary of everything needed to understand Chapter 2 and Chapter 3. The reader may skip parts or jump from one section to another as they see fit. The chapter is broadly divided as follows:

- Section 1.1 introduces all that is related to the notion of *cellular automata* (Section 1.1.1), central to the present part. After a few definitions on topology such as the one of *meager set* (Section 1.1.2), it describes several notions of attractors applied to cellular automata (Section 1.1.3), like the *omega-limit set*, and lays the mathematical groundwork evoked in the introduction of Part I that leads to the definition of the *generic limit set*.
- Section 1.2 introduces *one-dimensional subshifts* (Section 1.2.1), ties this back to cellular automata (Section 1.2.2), then shows how to see subshifts as graphs (Section 1.2.3). These notions, with immediate application in Chapter 3, are also extremely useful in Part II;
- Section 1.3 introduces two main elements of theoretical computer science: *Turing Machines* (Section 1.3.1), and as consequences the notions of *decidability* and *arithmetical complexity* (Section 1.3.2). Notions on Turing Machines are particularly used throughout this whole thesis.

1.1 Cellular automata and attractors

In all that follows, we name *alphabet* any *finite* set $\mathcal{A}$ of symbols.

1.1.1 Cellular automata

This subsection defines the one central discrete dynamical system of Part I: the cellular automaton.

Definition 1.1. A (one-dimensional) *cellular automaton* (CA) is a triple $(\mathcal{A}, F, r)$ where $\mathcal{A}$ is an alphabet and $F\colon \mathcal{A}^{2r+1} \to \mathcal{A}$ is a *local rule* of *radius* $r \in \mathbb{N}$. It defines a *global rule* $f\colon \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ by $f(x)(i) = F(x(i-r), x(i-r+1), \ldots, x(i+r))$ for all $x \in \mathcal{A}^{\mathbb{Z}}$ and $i \in \mathbb{Z}$. Most of the time, a cellular automaton will be denoted using $(\mathcal{A}, f)$ or simply f if the alphabet is clear.

This definition means that f simultaneously modifies each cell $x(i)$ of any "biinfinite word" $x \in \mathcal{A}^{\mathbb{Z}}$ written with alphabet $\mathcal{A}$, based only on its neighbors at distance at most r.

Example 1. Let f be the map so that for any $x \in \{0,1\}^{\mathbb{Z}}$, for any $i \in \mathbb{Z}$, we have $f(x)(i) = 0$ if and only if both $x(i-1)$ and $x(i+1)$ are 0, and $f(x)(i) = 1$ in any other case.

That application defines a cellular automaton on $\{0,1\}^{\mathbb{Z}}$, with the following local rule F, where the symbol 1 "spreads" to adjacent cells:

$$F\colon \{0,1\}^3 \to \{0,1\}$$
$$\begin{cases} F(000) &= 0 \\ F(w) &= 1 \quad \text{for any } w \in \{0,1\}^3 \setminus \{000\}. \end{cases}$$

As we have mentioned words improperly until now, it is of help to clarify what we mean by that in a general setting.

Definition 1.2. $\mathcal{A}^* := \cup_{n\in\mathbb{N}_0}\mathcal{A}^n$ is the set of all (finite) *words* that can be written with the alphabet $\mathcal{A}$. They are written $u = u_0 \dots u_{n-1}$ with $u_i \in \mathcal{A}$ for all $i \in \{0,\dots,n-1\}$, where $|u| := n$ is the *length* of u. When there is no confusion with a real number, ε represents the empty word, the only word with no symbol and of length 0.

We also define $\mathcal{A}^{\leq n} := \bigcup_{k\leq n}\mathcal{A}^k$ the set of words of length at most n.

For $0 \leq i \leq j$ and $u \in \mathcal{A}^*$, we define the *subword* $u_{[i,j]} := u_i u_{i+1} \dots u_{j-1} u_j$. We write $v \sqsubset u$ if v is a subword of u.

In all that follows, we use the notation x_i to refer to $x(i) \in \mathcal{A}$, for $x \in \mathcal{A}^{\mathbb{Z}}$ and $i \in \mathbb{Z}$. Any such x is called a *configuration*.

Remark 1. A configuration $x \in \mathcal{A}^{\mathbb{Z}}$ may also be referred to as a *biinfinite word*. However, in general the term "word" will only be used for finite words unless explicitly mentioned otherwise.

In all that follows, we endow $\mathcal{A}$ with the discrete topology, and $\mathcal{A}^{\mathbb{Z}}$ with the prodiscrete topology t_π – that is, the countable product of this discrete topology on each copy of $\mathcal{A}$. As such, $\mathcal{A}^{\mathbb{Z}}$ is compact.

Definition 1.3. Let $u \in \mathcal{A}^*$. The *cylinder* $[u]_i$ is the subset $\{x \in \mathcal{A}^{\mathbb{Z}} \mid x_{[i,i+|u|-1]} = u\}$ of $\mathcal{A}^{\mathbb{Z}}$.

We usually denote $[u] := [u]_0$.

If $W \subset \mathcal{A}^n$ is a set of words of some common length $n \in \mathbb{N}$, then we denote $[W]_i := \bigcup_{w\in W}[w]_i$ and $[W] := [W]_0$.

Property 1.4. *Cylinders form a basis of clopen*[1] *sets for the prodiscrete topology on the space* $\mathcal{A}^{\mathbb{Z}}$. *This topology is also induced by the metric*

$$d(x,y) := \inf_{n\in\mathbb{N}_0}\{2^{-n} \mid x_{[-n,n]} = y_{[-n,n]}\}.$$

A cellular automaton endowed with the profinite topology t_π forms a *dynamical system* $(\mathcal{A}^{\mathbb{Z}}, t_\pi, f)$: a space on which one studies the iterations of the function f and the behavior of the configurations. See Definition 7.1 for a general definition of dynamical systems, outside of the scope of this specific part.

[1] Sets that are both closed and open.

1.1.2 Notions of topology

The following subsection holds in general for Hausdorff compact[2] topological spaces. However, it will be restrained here to $X = \mathcal{A}^{\mathbb{Z}}$ for all intended purposes.

Definition 1.5. A subset $Y \subset X$ is *meager* (or *of first category*) if it is the union of countably many sets, with all of these sets having a closure with empty interior.

It is *comeager* (or *residual*) if its complement is meager; that is, if it is the intersection of countably many sets, so that all of these sets have a dense interior.

If $U \subset X$ is open and $Y \cap U$ is comeager in the relative topology on U, we say Y is *comeager in U*.

Example 2. On $\mathcal{A}^{\mathbb{Z}}$, we notably have the following:

- of course, the empty set is meager and $\mathcal{A}^{\mathbb{Z}}$ is comeager;
- since the closure of any singleton is itself and singletons in $\mathcal{A}^{\mathbb{Z}}$ are not isolated points, singletons have empty interior, hence they are meager sets;
- consequently, any countable subset of $\mathcal{A}^{\mathbb{Z}}$ is meager, as it is the union of countably many singletons;
- for any $u \in \mathcal{A}^* \setminus \{\varepsilon\}$, $[u]$ is neither meager not comeager.

Remark 2. Being meager represents being negligible in a topological sense. It is the topological equivalent of being of measure 0 in a measure space. Nevertheless, these two notions, even when they coexist in a topological measure space, are distinct. Consider for instance $\{0,1\}^{\mathbb{Z}}$ endowed with the topology t_π that is the product of the discrete topology on each copy of $\{0,1\}$. In it, any singleton is meager; however, if we use the trivial probabilistic measure μ_0 that gives weight 1 to the sole configuration $\dots 000 \dots$, we have a clear difference between meagerness and negligibility!

By the Baire Category Theorem applied to $\mathcal{A}^{\mathbb{Z}}$, as comeager sets are a countable intersection of dense sets, we have the following:

Property 1.6. *All comeager sets are dense in $\mathcal{A}^{\mathbb{Z}}$.*

Definition 1.7. We say Y has the *Baire property* if there is an open set $U \subset X$ such that $Y \Delta U$ is meager.

Remind that $Y \Delta U$ is the symmetric difference $(Y \setminus U) \cup (U \setminus Y)$.

Remark 3. Notice that any open set has the Baire property. Subsets of $\mathcal{A}^{\mathbb{Z}}$ with the Baire property form a σ-algebra, hence notably any Borel set in $\mathcal{A}^{\mathbb{Z}}$ has the Baire property.

Property 1.8. *If Y has the Baire property, then it is nonmeager if and only if it is comeager in some nonempty open set $U \subset X$.*

See for instance [Kec95, Section 8] for an overview of Baire-related notions.

[2] Hausdorff compact is equivalent to the French use of "compact" alone. The present author campaigns for these spaces to be called "French compact", to add to the general list of prefixes to the term compact.

1.1.3 Various notions of attractors

This subsection, just as the previous one, mostly holds within the general framework of dynamical systems (X, t, f) on Hausdorff compact spaces. However, for all purposes we will here restrict ourselves to cellular automata of the form $(\mathcal{A}^{\mathbb{Z}}, t_\pi, f)$ and consider $X = \mathcal{A}^{\mathbb{Z}}$ in what follows.

The notion of attractor of a dynamical system has sparked decades of debates about finding a common formal definition [Mil85]. Though it appears that several of them coexist, as detailed later in the present subsection, the first one a mathematician can reasonably come up with is the following:

Definition 1.9. The *omega limit* of $x \in X$ is the collection of all accumulation points in X of the orbit of x, defined by

$$\omega(x) := \bigcap_{N\in\mathbb{N}} \overline{\bigcup_{n\geq N} \{f^n(x)\}}.$$

Though the previous formula can be slightly off-putting at first, it represents a set that stems from intuition. Of course, the first notion of "attractor" in the orbit of a point x that comes to mind is the set of all the points that appear infinitely often in said orbit − that is, for any rank N, we can find a higher rank $n \geq N$ so that $f^n(x)$ is the desired point again. This is represented by the following formula:

$$\bigcap_{N\in\mathbb{N}} \bigcup_{n\geq N} \{f^n(x)\}.$$

However, using only that formula would cast aside all of the points that are not necessarily reached within the orbit of x, but still approached with any accuracy, which are very relevant when trying to define attractors. This is why the closure is necessary to define the omega limit $\omega(x)$.

Example 3. Reusing the cellular automaton "1 spreads over 0" from Example 1, consider any one configuration x of the form $\ldots 0011\ldots$: the formula without the closure yields an empty set, whereas the formula with it includes $\ldots 111\ldots$.

The set $\omega(x)$ is closed; and in $\mathcal{A}^{\mathbb{Z}}$ (where being compact and sequentially compact are one and the same) it is nonempty.

Now, the first envisioned ideas of an attractor for any subspace of X are given by the following definitions, notably used in [DG19]:

Definition 1.10. Let $Y \subseteq X$ be a subspace of X.

The *omega-limit set* of Y is

$$\Omega(Y) := \bigcap_{N\in\mathbb{N}} \overline{\bigcup_{n\geq N} \{f^n(Y)\}}.$$

The *asymptotic set* of Y is

$$\omega(Y) := \bigcup_{x\in Y} \omega(x).$$

Remark 4. The asymptotic set, in other articles, was also called ultimate set or approachable set. See notably [GR10] for a very detailed study of the omega-limit and asymptotic set of cellular automata.

Conceptually, $\omega(Y)$ is focused on the individual behavior of configurations from Y. It captures exactly the points that are approached infinitely often by the orbit – that is, the $\omega(y)$ – of any one point $y \in Y$. It is purposefully not closed; indeed, $\overline{\omega(Y)}$ also contains the accumulation points of these adherence configurations, which are not related to the individual behavior of any one configuration, see Example 5.

On the contrary, $\Omega(Y)$ is focused on the global behavior of the starting set Y. It is closed, and that is on purpose: it aims to capture the points that are approached with any accuracy even when "jumping" from the orbit of some y to the orbit of some y'. In some sense, these resulting points are all the ones that attract Y as a whole under the iteration of f – when not differentiating what comes from which orbit. Here, it makes sense to consider even the configurations of the closure of $\omega(Y)$, even if they are not in the adherence of any one orbit, as they are meaningful when the focus is on Y and not on its individual elements.

Property 1.11. *We have $\omega(Y) \subset \Omega(Y)$.*

Proof.

$$\begin{aligned}
\omega(Y) &= \bigcup_{x \in Y} \overline{\bigcap_{N \in \mathbb{N}} \bigcup_{n \geq N} \{f^n(y)\}} \\
&\subset \overline{\bigcup_{x \in Y} \bigcap_{N \in \mathbb{N}} \bigcup_{n \geq N} \{f^n(y)\}} \\
&\subset \overline{\bigcap_{N \in \mathbb{N}} \bigcup_{x \in Y} \bigcup_{n \geq N} \{f^n(y)\}} \\
&\subset \overline{\bigcap_{N \in \mathbb{N}} \overline{\bigcup_{x \in Y} \bigcup_{n \geq N} \{f^n(y)\}}} \\
&= \bigcap_{N \in \mathbb{N}} \overline{\bigcup_{x \in Y} \bigcup_{n \geq N} \{f^n(y)\}} \\
&= \Omega(Y)
\end{aligned}$$

□

The reverse inclusion may not hold, even when considering the closure of $\omega(Y)$ – see Example 4.

Example 4. Consider, once again, the same cellular automaton introduced in Example 1.

For any point x that is not $\ldots 000 \ldots$, we have $\omega(x) = \{\ldots 111 \ldots\}$. We consequently deduce that $\omega(X) = \{\ldots 000 \ldots, \ldots 111 \ldots\}$. However, $\Omega(X)$ is made of all the configurations that appear infinitely often in the system as a whole, which disqualifies any configuration with a finite isolated block of 1's, that does not appear anymore after some iteration. It can be shown that all other configurations are in it, and so it contains all $\ldots 0011 \ldots$ and $\ldots 1100 \ldots$ configurations in addition to the ones of $\omega(X)$.

Example 5. The following example of cellular automaton is presented in [GR10, Ex. 1]: consider $\mathcal{A} = \{\leftarrow, \rightarrow, L, R, \mathcal{W}, k\}$ with the following meanings and behavior:

- $\leftarrow$ and $\rightarrow$ represent a particle going to the left or the right;
- L and R are symbols left in the wake of a particle respectively going to the right or the left;

- $\mathcal{W}$ is a wall symbol, unmovable, on which particles "bounce back";
- k is a killer state: it appears whenever two particles collide and spreads in both directions, filling everything with k; they are only stopped in this spread by walls $\mathcal{W}$.

The resulting cellular automaton is made of "segments" (possibly infinite on one or, in specific cases, on both sides) of cells, each of them delimited by two wall symbols $\mathcal{W}$. For each segment, either it was containing only one particle at the start (and no killer state k) and becomes periodic after some point in time, with one particle surrounded by L states on its left and R states on its right; or it ends up full of cells with k.

Notably, considering a set Y made of a sequence of configurations so that each of them has a central segment $\mathcal{W}L\dots L \to R\dots R\mathcal{W}$ where $\to$ is at position 0 and the two $\mathcal{W}$'s are at positions $-\ell$ and ℓ, with growing ℓ, we have the configuration $\dots LL \to RR\dots$ in $\overline{\omega(Y)}$, with infinitely many L's on the left and infinitely many R's on the right. Yet it is in no $\omega(x)$ for any $x \in Y$, and so $\omega(Y)$ is not closed.

We want to build a notion that has the best of both worlds: something finer than an omega-limit set, but closed because that is a nice property. As such, it will necessarily be a compromise between the asymptotic set (configurations-focused) and the omega-limit set (set-focused); as we see in Property 1.15, it will end up being some $\overline{\omega(Y)}$ for a very well-chosen Y. That careful choice stems from wanting to ignore any negligible set of starting configurations. For instance, in Example 4, the configuration $\dots 111\dots$ is in $\Omega(X)$ and $\omega(X)$ but only attracts itself, as f acts as the identity on it. In trying to consider an attractor that represents the long-term behavior of a system overall, we want to not take into account such singularities.

To do so, we rely on the notion of meagerness, as in Definition 1.5, that allows to formalize a topological negligibility. In order to have a counterpart to the notion of "attractor that attracts a given set Y" on which both the asymptotic and the omega-limit sets are based in their own way (focused on its local or its global behavior), we change our point of view and define the set of "all points that a given closed set A attracts".

Definition 1.12. Given a closed set $A \subset X$, we define the *basin* (or sometimes *realm*) *of attraction* $\mathcal{D}(A)$ of A as

$$\mathcal{D}(A) := \{x \in X \mid \omega(x) \subset A\}.$$

We say that A is a (topological) *attractor* if $\mathcal{D}(A)$ is nonmeager, and a (topological) *generic attractor* if $\mathcal{D}(A)$ is comeager.

Any basin of attraction $\mathcal{D}(A)$ has the Baire property [DG19, proof of Prop. 3.12], defined in Definition 1.7. Philosophically speaking, this means that any attractor A is somewhat generic up to some "rescaling process" – the restriction to a well-chosen closed, stable subset in which $\mathcal{D}(A)$ is comeager.

In what follows, we consequently focus on generic attractors only. They describe the long-term behavior of most points' orbits of a dynamical system X, with possible eviction of "singular" points – since the basin of attraction only has to be comeager, and not necessarily be the whole space. However, among the various generic attractors that may exist in a dynamical system, how can we single out one of them such that:

- it does not contain any point that attracts only a meager portion of the whole space;

- it still contains $\omega(x)$ for any meaningful x?

We define the following:

Definition 1.13. Let $\widetilde{\omega}(f)$ be the intersection of all generic attractors of a cellular automaton. It is called the *generic limit set* (GLS).

It is mostly easy to understand that $\widetilde{\omega}(f)$ always verifies the following properties:

Property 1.14.
- *$\widetilde{\omega}(f)$ is a closed set;*
- *the basin of $\widetilde{\omega}(f)$ is comeager;*
- *$\widetilde{\omega}(f)$ is a generic attractor, and the smallest of them inclusion-wise;*
- *$\widetilde{\omega}(f)$ is nonempty.*

Proof. Property 1 is because any intersection of closed sets is closed; property 3 results from properties 1 and 2; property 4 holds because of property 2 and the fact that the empty set has no comeager basin.

Property 2, the only nontrivial statement, is due to the fact that $\mathcal{D}(\widetilde{\omega}(f))$ is the intersection D of the basins of all generic attractors. Indeed,

$$\begin{aligned} x \in \mathcal{D}(\widetilde{\omega}(f)) &\Leftrightarrow \omega(x) \subset \widetilde{\omega}(f) \\ &\Leftrightarrow \forall A \text{ generic attractor}, \omega(x) \subset A \\ &\Leftrightarrow \forall A \text{ generic attractor}, x \in \mathcal{D}(A) \\ &\Leftrightarrow x \in D. \end{aligned}$$

It remains to show that this intersection is countable. Here, $X = \mathcal{A}^{\mathbb{Z}}$ is a compact space that can be given a metric (see for instance Example 12). Any compact metric space has a countable basis – that is, countably many closed sets C_i (for i in some I) so that any other closed set can be written as a countable intersection of some of them. As such, any basin of a generic attractor, being comeager and so a closed set, can be written as a countable intersection of C_i's. Consequently, D can be written as a countable (subset of I) intersection of C_i's. Therefore $D = \mathcal{D}(\widetilde{\omega}(f))$ is comeager. $\square$

This notion of generic limit set corresponds to what we needed, because by definition its basin of attraction is the smallest it can be while still being comeager. Besides, consider $\omega(\mathcal{D}(\widetilde{\omega}(f))) = \cup_{x\in\mathcal{D}(\widetilde{\omega}(f))}\omega(x)$: it is a subset S of $\widetilde{\omega}(f)$ by definition of $\mathcal{D}(\widetilde{\omega}(f))$. When taking the closure, we consequently obtain an attractor $\overline{S}$ that is included in $\widetilde{\omega}(f)$. Moreover, $\mathcal{D}(\overline{S}) \supset \{x \in X \mid \omega(x) \subset S\} \supset \mathcal{D}(\widetilde{\omega}(f))$ so $\mathcal{D}(\overline{S})$ is comeager. Since $\widetilde{\omega}(f)$ is the smallest generic attractor whose basin is comeager, we have that

Property 1.15. $\overline{\omega(\mathcal{D}(\widetilde{\omega}(f)))} = \widetilde{\omega}(f)$.

Therefore, $\widetilde{\omega}(f)$ does attract most of the space the way we wanted.

Example 6. We reuse Example 1, the CA with "spreading 1" and alphabet $\{0, 1\}$. Considering that for any subset $Y \subset \{0,1\}^{\mathbb{Z}} \setminus \{\dots 000 \dots\}$ is so that $\omega(Y) = \{\dots 111 \dots\}$, and using that the GLS is nonempty and Property 1.15, we have $\widetilde{\omega}(f) = \{\dots 111 \dots\}$.

Example 7. Consider the cellular automaton on the alphabet $\{0,1\}$ with global rule $f(x)_i = \max(x_{i+1}, x_{i+2})$. It is slightly different from the previous example: it is of the form "spreading 1", but there is also some shift to the left operated by f. As a consequence, the generic limit set is different from the one of the previous example. It is clear that no word of the form 10^n1 appears in the generic limit set, as they all disappear from any starting configuration after $\lceil n/2 \rceil$ iterations of f. However, any starting configuration with 0^{2n+k} as a subword contributes to the appearance of 0^k at iteration n. This goes to show that, considering configurations containing 0^{2n+k} as a subword don't form a meager set for any n and any k, that any word of the form 0^k *does* appear in some configuration of the generic limit set. Similarly, $1^{k_1}0^{k_2}$ (and $0^{k_2}1^{k_1}$) can still be found in a nonmeager set of configurations after any number of iterations. From this, we conclude that $\widetilde{\omega}(f)$ is made of $\ldots 111 \ldots$, $\ldots 000 \ldots$, and all configurations of the form $\ldots 0011 \ldots$ and $\ldots 1100 \ldots$.

Overall, a combinatorial characterization of the words that appear in configurations of the generic limit sets for cellular automata has been obtained by Törmä in [Tör20] and will be of great use in this entire part.

Lemma 1.16 (Lemma 2 in [Tör20])**.** *Let f be a CA on $\mathcal{A}^{\mathbb{Z}}$. A word $s \in \mathcal{A}^*$ occurs in $\widetilde{\omega}(f)$ if and only if there exists a word $v \in \mathcal{A}^*$ and $i \in \mathbb{Z}$ such that for all $u, w \in \mathcal{A}^*$ there exist infinitely many $t \in \mathbb{N}$ with $f^t([uvw]_{i-|u|}) \cap [s] \neq \varnothing$.*

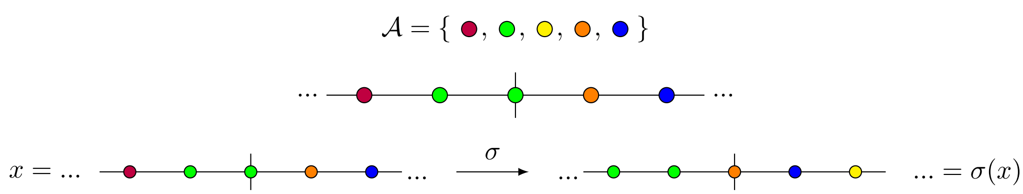
We say that v enables *s for f.*

1.2 One-dimensional subshifts

1.2.1 Main notions

Let σ denote the (left) *shift map* $\sigma \colon \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$. It is an homeomorphism of $\mathcal{A}^Z$ defined by $\sigma(x)_i = x_{i-1}$, so that $\sigma(x)$ is simply x shifted one coordinate to the left.

Example 8. In the picture that follows, we illustrate an alphabet $\mathcal{A}$, a local, origin-based window on a specific configuration from $\mathcal{A}^{\mathbb{Z}}$ (with the vertical bar representing the origin), and the effect of the shift map σ.



$(\mathcal{A}^{\mathbb{Z}}, t_\pi, \sigma)$ is a dynamical system called the *full shift* for the alphabet $\mathcal{A}$.

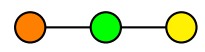
Definition 1.17. A *subshift* X is a σ-invariant closed subset of $\mathcal{A}^{\mathbb{Z}}$.

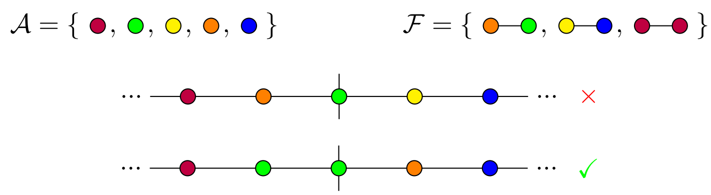
In the general theory of dynamical systems, a subshift is simply a subsystem of the full shift.

Property 1.18. *Any subshift X can be described by a set $\mathcal{F} \subset \mathcal{A}^*$ of words called* forbidden patterns*, meaning the following holds:*

$$X = \mathcal{A}^{\mathbb{Z}} \setminus \left(\bigcup_{u \in \mathcal{F}} \bigcup_{i \in \mathbb{Z}} [u]_i \right).$$

It is often denoted $X_{\mathcal{F}}$ instead of merely X.

Remark 5. In simple terms, $X_{\mathcal{F}}$ is the set of all configurations that do not contain any word from $\mathcal{F}$. Several sets of forbidden patterns can define the same subshift: for instance, in the following example, forbidding an additional  would be of no consequence on the allowed configurations, as it is already implied by an existing forbidden pattern.



Definition 1.19. A *morphism* $f\colon X \to Y$ between two subshifts is a continuous function with $f \circ \sigma = \sigma \circ f$.

If f is surjective, it is called a *factor map* and Y is said to be a *factor* of X. If f is bijective, the two systems are said to be *conjugate* – this can be denoted $X \cong Y$.

As it often is in mathematics, two conjugated subshifts are often considered to be more or less “the same” object. *Conjugacy invariants* are properties so that if X has property $\mathcal{P}$ and is conjugated to Y, then Y has property $\mathcal{P}$.

Definition 1.20. If a subshift $X \subset \mathcal{A}^{\mathbb{Z}}$ is so that $X \cong X_{\mathcal{F}}$ with $\mathcal{F}$ that is finite, then X is called a *subshift of finite type, or SFT for short.*

If a subshift is a factor of an SFT, then it is called a *sofic subshift.*

If a subshift $X \subset \mathcal{A}^{\mathbb{Z}}$ is so that $X \cong X_{\mathcal{F}}$ with $\mathcal{F}$ that can be enumerated by a Turing Machine, then X is called an *effective subshift.*

See Definition 1.36 for the definition of a Turing Machine. The following property is an easy consequence of the definitions:

Property 1.21. *All the notions from the previous definition are conjugacy invariants.*

Property 1.22. *We have the following successive implications, for a subshift X:*

$$X \textit{ is of finite type} \Rightarrow X \textit{ is sofic} \Rightarrow X \textit{ is effective.}$$

Example 9. Let $\{0,1\}$ be the alphabet $\mathcal{A}$ for all the examples that follow.

$X_{\{11\}}$ is an example of SFT, where the only restriction is that two 1's must not follow each other.

An example of sofic subshift is $X_{\{10^n1 \mid n\in\mathbb{N}_0\}}$, traditionally denoted as $X_{\leq 1}$, which contains all configurations that contain at most one 1. It cannot be written using a finite number of forbidden patterns, as it must forbid any distance between two 1's. However, it is sofic, because it is a factor of the SFT $X_{\{\to\leftarrow,\leftarrow\to,1\to,\leftarrow 1,11\}}$ that is a subshift on $\{\leftarrow,1,\to\}$, where 1 is projected on 1, and $\leftarrow$ and $\to$ are projected on 0.

An effective subshift that is not sofic would be $X_{\{10^n1 \mid n\in Halt\}}$, where $Halt$ is, when enumerating all Turing Machines, the set of all of them that halt on the empty input. A short proof of this subshift not being sofic is that its language is Σ^0_1 (see Section 1.3.2) while sofic subshifts have a regular, hence computable language.

Note that some subshifts may not even be effective, such as $X_{\{10^n1 \mid n\notin Halt\}}$. Indeed, if it were to be conjugated to another subshift $X_{\mathcal{F}}$ with $\mathcal{F}$ that can be enumerated by a Turing Machine, then it would mean the language of X is Σ^0_1 (see Section 1.3.2 again), which is not the case.

1.2.2 Ties with cellular automata

As we have introduced our two main tools for the present part, subshifts and cellular automata, we present two unrelated results providing ties between these two classes of dynamical systems.

Theorem 1 (Curtis-Hedlund-Lyndon, 1969)**.** *For a given alphabet $\mathcal{A}$, $(\mathcal{A}, f)$ is a cellular automaton if and only if f is an endomorphism of the full shift $(\mathcal{A}^{\mathbb{Z}}, t_\pi, \sigma)$.*

Proposition 1.23 (Prop. 4.11 from [DG19])**.** *Let f be a CA. Then the generic limit set $\widetilde{\omega}(f)$ is a nonempty f-invariant subshift.*

Proof. Since it is a subset of the full shift $\mathcal{A}^{\mathbb{Z}}$, it is enough to prove that $\widetilde{\omega}(f)$ is closed under the topology and the action of the left shift σ, and that it is f-invariant. Its nonemptiness and the fact that is is closed under the topology are already known from Property 1.14.

We prove that $\widetilde{\omega}(f)$ is closed under the action of the shift σ. Consider $y \in \widetilde{\omega}(f)$: y belongs to some $\omega(x)$ where $x \in \mathcal{D}(\widetilde{\omega}(f))$. Then, noticing that f and σ commute due to Definition 1.1, and that σ is bijective and continuous, we have $\sigma(y) \in \sigma(\omega(x)) = \omega(\sigma(x))$. Now it is straightforward to see that $\sigma(x) \in \mathcal{D}(\widetilde{\omega}(f))$, and so $\sigma(y) \in \widetilde{\omega}(f)$. Similarly, $\sigma^{-1}(y) \in \widetilde{\omega}(f)$, and so $\sigma(\widetilde{\omega}(f)) = \widetilde{\omega}(f)$.

Now, $y \in \widetilde{\omega}(f)$ if and only if it belongs to some $\omega(x)$ for a given $x \in \mathcal{D}(\widetilde{\omega}(f))$. But y belongs to $\omega(x)$ if and only if $f(y)$ does, due to the definition of $\omega(x)$; and so $y \in \widetilde{\omega}(f)$ if and only if $f(y) \in \widetilde{\omega}(f)$. As such, $\widetilde{\omega}(f)$ is f-invariant. □

1.2.3 One-dimensional subshifts seen through graphs

In this subsection, we associate graphs to one-dimensional subshifts: with this, several tools and some understanding from graph theory can be applied to subshifts.

First, we need a few notions and notations to manipulate forbidden patterns of a given size.

Definition 1.24. Let $X \subset \mathcal{A}^{\mathbb{Z}}$ be a subshift.

The *language* $\mathcal{L}(X) := \{v \in \mathcal{A}^* \mid \exists x \in X, v \sqsubset x\}$ of a subshift X is the (countable) set of all subwords of configurations in X.

We also denote $\mathcal{L}_n(X) := \mathcal{L}(X) \cap \mathcal{A}^n$ the set of subwords of length n in configurations of X.

Definition 1.25. Let $X_{\mathcal{F}} \subset \mathcal{A}^{\mathbb{Z}}$ be an SFT.

If $\mathcal{F} \subset \mathcal{A}^{\leq n}$ but $\mathcal{F} \not\subset \mathcal{A}^{\leq n-1}$, then we say that $X_{\mathcal{F}}$ has *window size* n.

A SFT of window size 2 is said to be *nearest-neighbor*.

Notice that a SFT of window size 1 is conjugate to a full shift on a smaller alphabet.

The following definitions are powerful properties of some subshifts, that have been extensively studied – see for instance [Kůr03] for the general theory of dynamical systems. They are, as a whole, called *mixing properties*, and in subshifts represent more or less strong ways to mix together patterns into a full configuration. More mixing properties exist in two dimensions and more [BPS10, BMP18]. Here, in one dimension, we focus on the following two:

Definition 1.26. Let X be a subshift.

We say X is *transitive* if for all $u, v \in \mathcal{L}(X)$, there exists $w \in \mathcal{L}(X)$ so that $uwv \in \mathcal{L}(X)$.

We say X is *mixing* if for all $u, v \in \mathcal{L}(X)$, for any $n \in \mathbb{N}_0$ large enough possibly depending on u and v, there exists $w \in \mathcal{L}_n(X)$ so that $uwv \in \mathcal{L}(X)$.

If X is a mixing SFT, then there exists a uniform $n \in \mathbb{N}_0$, called its *mixing distance*, that works for all u, v.

The last part of the definition holds because of the bounded size of forbidden patterns in SFTs – it can be seen easily when considered from a graph-related point of view, see Property 1.31.

The above definition gives rise to the obvious property

Property 1.27. *X is mixing $\Rightarrow$ X is transitive.*

Example 10. The subshift $X_{\{00\}} \subset \{0,1\}^{\mathbb{Z}}$ is mixing: two valid words can be glued together by adding any nonzero number of 1's. Consequently, its mixing distance is 1.

The subshift $X_{\{00,11\}} \subset \{0,1\}^{\mathbb{Z}}$ is made of exactly two configurations, the alternations of 0 and 1 with either 0 or 1 at the origin. It is transitive, because any valid word can be glued to any other – all of them are alternations of 0 and 1, after all – but it is not mixing, as only specific distances work for the gluing process.

The subshift $X_{\leq 1}$ already mentioned in example Example 9 is not transitive since two 1's, that are valid words in an of themselves, cannot be glued together.

Now, to dive into elements of graph theory, as mentioned at the beginning of this subsection or in [LM95, Chapter 2], we define:

Definition 1.28. Let $X \subset \mathcal{A}^{\mathbb{Z}}$ be a one-dimensional SFT.

The *Rauzy graph of order M of X* is the directed graph $\mathcal{G}_M(X) = (\mathcal{V}, \vec{E})$ with:

- $\mathcal{V} = \mathcal{L}_M(X)$;
- $(u_1 \dots u_M, u_2 \dots u_{M+1}) \in \vec{E}$ for $u_1, \dots, u_{M+1} \in \mathcal{A}$ if and only if $u_1 \dots u_{M+1} \in \mathcal{L}_{M+1}(X)$
- no other couple of vertices belongs to $\vec{E}$.

Considering its vertices correspond to elements of the language of X, that can by definition be extended to full configurations, no Rauzy graph contains any *stranded vertex* – a vertex with in-degree or out-degree 0.

In all that follows, if a Rauzy graph is mentioned without any precision or denoted as $\widetilde{\mathcal{G}}(X)$ for a SFT X, it is implied that it is its Rauzy graph $\mathcal{G}_{M-1}(X)$ of order $M-1$, with M the size of its largest forbidden pattern.

Furthermore, the following results allow to focus even more precisely on nearest-neighbor SFTs and their Rauzy graphs. More precisely, they imply that most proofs can focus on combinatorics over graphs with no stranded vertex to describe somewhat naturally all one-dimensional SFTs up to conjugacy. Proofs can be found in [LM95, Chapter 2] for instance.

Property 1.29. *If X is nearest-neighbor, then up to a renaming of the symbols, $\widetilde{\mathcal{G}}(X) = (\mathcal{V}, \vec{E})$ is the unique graph with no stranded vertex so that X can be described with:*

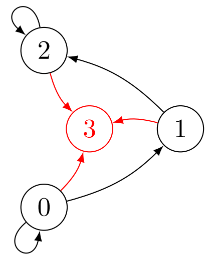
- *$\mathcal{V} = \mathcal{A}$;*
- *for any $a, b \in \mathcal{A}$, $ab \in \mathcal{F}$ if and only if $(a,b) \notin \vec{E}$.*

Property 1.30. *Any SFT is conjugated to a nearest-neighbor SFT.*

The previous two properties are of great use, because they mean that most proofs can focus on combinatorics over graphs with no stranded vertex to describe exactly all one-dimensional SFTs up to conjugacy.

Any directed graph is composed of one or several *strongly connected components* (*SCC* for short)[3]. If a graph has several SCCs, it can also contain *transient vertices*: vertices with no path from themselves to themselves, that form a path from one SCC to another.

Example 11. The subshifts $X = X_{\{10,21,20,11,30,31,32,33\}} \subset \{0,1,2,3\}^{\mathbb{Z}}$ and $Y = Y_{\{10,21,20,11\}} \subset \{0,1,2\}^{\mathbb{Z}}$ are the same SFT.



They have the same Rauzy graph of order 1 represented above in black, made of two SCCs $\{0\}$ and $\{2\}$, and one transient vertex 1. Notice how adding the vertex and edges in red could still fit with a description of the adjacency rules of X, but are not part of its Rauzy graph – the word 23 is locally authorized but not in $\mathcal{L}(X)$.

Now, we introduce an easy way to see the transitivity or mixing property of an SFT, the proof of which is omitted as it stems naturally from the definition of the Rauzy graph of fitting order:

Property 1.31. *Let X be a SFT.*

$$X \text{ is transitive} \Leftrightarrow \widetilde{\mathcal{G}}(X) \text{ is made of exactly one SCC.}$$

$$X \text{ is mixing} \Leftrightarrow \widetilde{\mathcal{G}}(X) \text{ is made of one SCC and is an aperiodic graph.}$$

where an aperiodic graph is a graph so that the GCD of its cycles' length is 1.

The following property can be proved using the basic definitions or the above graph-theoretical equivalents.

Property 1.32. *Being transitive and being mixing are conjugacy invariants.*

The use of graph theory does not hold as well for subshifts in general. However, the notion of Rauzy graph still makes sense, and allows us to introduce approximations of subshifts.

Definition 1.33. The *SFT approximation of width n* of X is the SFT $\mathcal{S}_n(X)$ defined by the forbidden patterns $\mathcal{A}^n \setminus \mathcal{L}_n(X)$. It corresponds to the Rauzy graph $\mathcal{G}_n(X)$.

Note that $X = \cap_n \mathcal{S}_n(X)$.

Notice that any SFT approximation $\mathcal{S}_n(X)$ of a SFT X yield that exact SFT as long as n is big enough (that is, bigger than the SFT's window size).

Definition 1.34. Let X be a subshift.

We say that X is *chain transitive* (resp. *chain mixing*) if every $\mathcal{S}_n(X)$ is transitive (resp. mixing).

[3]The difference with the usual notion of connected components is that for SCCs, one has to follow the directed edges.

See [Aki93, pp. 66 and 175] for definitions in the context of general dynamical systems, equivalent to these ones in the special case of subshifts. Chain transitivity, notably applied to subshifts, is also mentioned in the classic reference [Kůr03, Def. 2.5 and Prop. 3.62].

Of course, we have the following results:

Property 1.35. *Let X be a subshift.*

X is transitive (resp. mixing) $\Rightarrow$ X is chain transitive (resp. chain mixing).

If X is a SFT, then the reverse implication also holds.

1.3 Notions of arithmetical complexity

1.3.1 Turing Machines and decidability

From a certain point of view, the simplest theoretical way of representing a computer is to use a tape containing data, that can be rewritten according to certain rules, and depending on the current content of the tape and some inner state. This notion is the one of Turing Machines (occasionally abbreviated TM), and it is formalized below.

Definition 1.36. A *Turing Machine* is a 5-tuple $(Q, \mathcal{A}, q_i, q_f, \delta)$, where

- Q is a finite set of states with initial state q_i and final state q_f;
- $\mathcal{A}$ is a finite alphabet containing some "blank" symbol \$;
- δ is a transition rule from $\mathcal{A} \times Q$ to $\mathcal{A} \times Q \times \{\leftarrow, \rightarrow\}$.

The machine is represented by a read/write head positioned somewhere on an infinite discrete tape, that reads letters written with $\mathcal{A}$ on the tape and overwrites them according to the transition rule δ and its current state $q \in Q$. It then moves to the left ($\leftarrow$) or the right ($\rightarrow$).

It starts in state q_i with its head at some position 0, and is given an *input* in $\mathcal{A}^*$ (completed with blank symbols \$ to fill the infinite tape) or $\mathcal{A}^{\mathbb{Z}}$ (effectively enumerable input[4].).

If the machine ends up in q_f, it *halts* and outputs the content of its tape. Else, it does not halt and never outputs anything. See Fig. 1.1.

Remark 6.

- Other ways to represent Turing Machines (using several tapes or several halting states, for instance) exist. All of them are equivalent.
- It is commonly accepted that Turing Machines have the computational power of computers (Church-Turing thesis).
- A Turing Machine $\mathcal{M}$, having a finite number of states, finite alphabet, and consequently a finitely describable transition rule, can be encoded as a finite word written as $\langle \mathcal{M} \rangle$. Then, this "code" can be used as input by another Turing Machine.

[4]Some infinite configurations are not accepted by Turing Machines, for instance because they would contradict Theorem 2; the ones that are are called the effectively enumerable inputs.

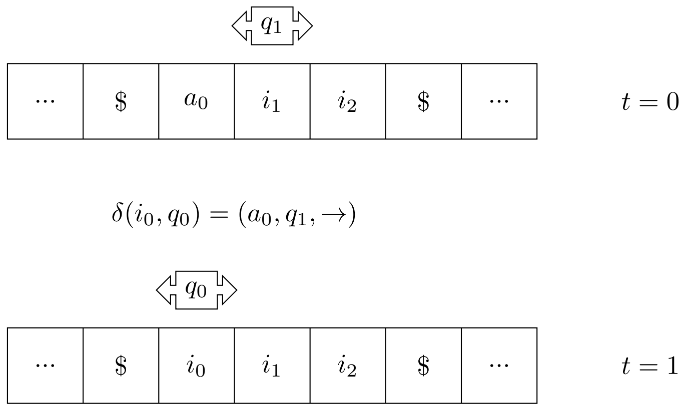


Figure 1.1: A representation of a Turing Machine, performing the transition $\delta(q_0, i_0) = (q_1, a_0, \rightarrow)$. Time goes upward.

Note that a Turing Machine's transition rules can be encoded in the local rules of a cellular automaton, if the latter stores at all times on a cell the position of the head and the current state of the TM. However, this requires the correct formatting of the cellular automaton into finite segments, each dedicated to the computation of a given copy of the machine. Only by forcing these delimitations – at least in most starting configurations, which is fine since the generic limit set can neglect a meager set of them – can we ensure that no two heads of machines risk colliding, and that each one of them has its own clean, finite starting tape. An important consequence is that a cellular automaton is able to run space-bound copies of any Turing Machine: this will be of use in Section 3.1.

Definition 1.37. A subset S of $\mathbb{N}$ is said to be *computable* if there exists a Turing Machine that, taking any number as input, halts in any case and answers Yes if and only if the number belongs to S, and No if and only if it does not.

A computable subset of $\mathbb{N}$ is also said to be Σ_0^0 or Π_0^0.

The definition of computability is *extended to any subset of any countable set* using a reasonable, agreed upon bijection with $\mathbb{N}$. In most of this PhD, we reason on sets of words $\mathcal{A}^*$ for some alphabet $\mathcal{A}$; the bijection is the numbering of the words, lexicographically for a given length, then with increasing length.

Definition 1.38. A *decision problem* for property $\mathcal{P}$ on inputs from $S \subset \mathcal{A}^*$ is the set $\{w \in S \mid w \text{ has property } \mathcal{P}\} \subset \mathcal{A}^*$

In general, it is defined as a question about property $\mathcal{P}$ on a set of finitely-encodable inputs that expects a Yes/No answer for each of them.

A decision problem is said to be *decidable* if it is a computable set. Said otherwise, there exists a Turing Machine $\mathcal{U}$ such that for any input w of the decision problem, $\mathcal{U}$ accepts w if the answer to w having property $\mathcal{P}$ is Yes, and rejects w if the answer to w is No – thus halting in any case.

Else, the decision problem is said to be *undecidable*.

A widely known theorem is the following:

Definition 1.39. The Halting Problem is the decision problem taking as input a Turing Machine $\mathcal{M}$ encoded as a word $\langle\mathcal{M}\rangle$, and an input w for that machine, and answering Yes if $\mathcal{M}$ halts on w, No if it does not.

Theorem 2 (Undecidability of the Halting Problem)**.** *The Halting Problem is undecidable.*

Numerous proofs of this fact can be found in the literature[5].

Remark 7. The undecidability of the Halting Problem is fundamental among decision problems. Indeed, most proofs of undecidability can be reduced (see Definition 1.42) to the one of the Halting Problem.

1.3.2 Complexity

Definition 1.40. A first-order logical formula with one free variable on $\mathbb{N}$ is said to be computable if and only if the subset on which it holds true is itself computable.

Definition 1.41. Let φ be a first-order logical formula on $\mathbb{N}$ with one free variable k. We define the following recursively:

- if φ is computable, then φ is said to be Σ_0^0 and Π_0^0;
- if φ is logically equivalent to some $\exists k_1 \dots \exists k_m \psi$ where ψ is a Π_n^0 logical formula with free variables $k_1, \dots, k_m$ and k, then φ is Σ_{n+1}^0;
- similarly, if φ is logically equivalent to some $\forall k_1 \dots \forall k_m \psi$ where ψ is a Σ_n^0 logical formula with free variables $k_1, \dots, k_m$ and k, then φ is Π_{n+1}^0

We also define $\Delta_n^0 = \Pi_n^0 \cap \Sigma_n^0$ as the set of all computable formulas.

This is called the *arithmetical hierarchy.*

Though the previous definition holds on logical formulas, it can also be defined on sets directly, by the use of successive caps and cups: for instance, a Σ_1^0 set is the union, over a computable set, of countably many computable sets. Indeed, "φ holds on $\cup_{k_1} U_{k_1}$" means "$\exists k_1, \varphi$ holds on U_{k_1}" and so this fits with the definition through quantifiers on logical formulas.

As mentioned previously, all these definitions can be extended to any countable set using a decent bijection with $\mathbb{N}$.

Definition 1.42. Given two Σ_n^0 (resp. Π_n^0) sets A and B, we say that B can be *(many-one) reduced* to A if there exists a Turing Machine that, when given a black box procedure able to tell whether any given element belongs to A, can compute B using the black box procedure only once at the end of its process.

Fundamentally, B can be reduced to A if the cost of algorithmically describing B is at most the cost of describing A – that is, A is "harder" than B. The presence of a black box procedure in the definition is of use because the actual difficulty of computing A is ignored, only the relative difficulty of computing B from A is studied.

[5]As an example, a short poetic proof by G. K. Pullum can be found at the following address: http://www.lel.ed.ac.uk/ gpullum/loopsnoop.html.

Remark 8. Notice how the black box procedure (called an oracle) is restrained to only one use at the very end of the algorithm/Turing Machine's process. If that restriction is ignored, we obtain the broader definition of Turing reduction: for instance $Halt$, the set of all Turing Machines halting on the empty set, is Turing-reducible but not (many-one) reducible to its complement. Indeed, if the output of the black box procedure can be altered afterwards, its Yes and No outputs (answering whether an input belongs to a set of not) can be swapped; and so any set is Turing-reducible to its complement. However, a many-one reduction of $Halt$ to its complement would allow said complement to be Σ_1^0, just as $Halt$ is; and so $Halt$ would be Δ_1^0, which is decidable, contradicting Theorem 2.

As before, the same definition holds for logical formulas.

Property 1.43. *Suppose B can be reduced to A.*
If B is undecidable, then so is A.
If A is decidable, then so is B.

Definition 1.44. A set A is Σ_n*-hard* if any Σ_n set can be reduced to A; it is Σ_n*-complete* if it is Σ_n and Σ_n-hard.
The same definitions hold for Π_n and Δ_n.

In all that follows, we say that a subshift is of a given arithmetical complexity if its language is.

Chapter 2

Language Hardness and Complexity Bounds

This chapter, based on the first half of [ENT22], is dedicated to complexity and structural results about generic limit sets of cellular automata.

Section 2.1 goes through a brief history about what is known on arithmetical complexity of generic limit sets, and makes some parallels with known complexities for omega-limit sets (Definition 1.10) and μ-limit sets (Definition 2.1).

Section 2.2 details two new bounds on the complexity of generic limit sets: they are Π_2^0 if the cellular automaton is *inclusion-minimal* (Section 2.2.1), and Σ_1^0 if the cellular automaton has *equicontinuity points* (Section 2.2.2).

Finally, Section 2.3 proves additional results related to generic limit sets and *shift-minimality*.

2.1 Brief state of the art

To give some perspective on the following newfound results about generic limit sets, we introduce an additional notion of attractor, related to measure-theory, on which various results have been obtained [BDS10].

Definition 2.1. Let μ be a measure on $\mathcal{A}^{\mathbb{Z}}$ and f a CA. The *μ-limit set* $\Lambda_\mu(f)$ of f is the subshift whose forbidden patterns are exactly the words $u \in \mathcal{A}^*$ such that

$$\lim_{n\to\infty} \mu(f^{-n}([u])) = 0.$$

Remark 9. There exists a measure-theoretic equivalent of the generic limit set, called the likely limit set [Mil85] – the closed set with the smallest possible basin of attraction of measure 1. The μ-limit set, however, works differently: it is made of all words whose frequency of appearance (among μ-weighted configurations) does not vanish as time goes to infinity.

Consequently, a word can appear with increased rarity without ever disappearing entirely from the configurations' orbits, and not be in the μ-limit set; this is not the case for the generic limit set.

In Table 2.1, inspired by the table in [BDP$^+$15, Section 7], we recapitulate the state of the art so far regarding several properties of the three commonly-considered CA attractors: the omega-limit set, the μ-limit set where μ is the uniform Bernoulli measure, and the generic limit set.

Among the properties studied, one is about equicontinuity points. To introduce this briefly, an equicontinuity point x is an element of $\mathcal{A}^{\mathbb{Z}}$ so that configurations close to x have orbits that stay close to x's. A proper definition is given in Definition 2.8.

Table 2.1: Comparison of computability properties of different limit sets. The measure μ is the fully supported Bernoulli probability measure.

Problem or property	**Omega-limit set**	**μ-limit set**	**Generic limit set**
Being a singleton	Σ_1^0-complete [Kar92]	Π_3^0-complete [BDP+15, Th. 5.7]	Σ_2^0-complete [Tör21, Th. 8]
Any non-trivial property	Σ_1^0-hard [Kar94]	Π_3^0-hard [BDP+15, Th. 5.2]	undecidable [Del21, Th. 4.1]
Worst-case language	Π_1^0-complete [Hur87, Th. 4]	Σ_3^0-complete [BDP+15, Th. 4.4]	Σ_3^0-complete [Tör20, Th. 1]
Worst-case language when f acts as a shift map on the attractor	computable (is a SFT) [Taa07, Th. 1]	Σ_3^0-complete [BDP+15, Th. 4.4]	Π_2^0-complete: Corollary 3.15, Proposition 2.7 and Corollary 3.16
Worst-case language when f has equicontinuity points	Π_1^0-complete [Hur87, Th. 4]	Σ_1^0-complete: [BDP+15, Th. 4.2], [BDP+15, Prop. 4.1] and Remark 10	Σ_1^0-complete: Proposition 2.11, Lemma 3.3 and Theorem 3

The Π_1^0-completeness results on omega-limit sets – that is, the fact that the bound is tight – come from [Hur87, Theorem 4] and its proof. Hurd constructs a CA that simulates copies of a Turing Machine on disjoint tapes and has a Π_1^0-complete omega-limit language. Moreover, the tapes cannot be extended or destroyed, and no information can pass from one tape to another, so a short tape bordered by two other tapes forms a blocking word. Hence, due to a common result stated here in Proposition 2.10, the CA built that way admits equicontinuity points.

The Σ_3^0-completeness results on μ-limit sets – that is, once again, the tightness of the corresponding bounds – follow from [BDP+15, Theorem 4.4]. Its authors construct a CA that has a Σ_3^0-complete μ-limit set Furthermore, f acts on it as the identity (a trivial shift map).

Several bounds in complexity for generic limit sets were already known from [Tör20]; we mention the following, to give some perspective on the next results:

Proposition 2.2 (Th. 1 from [Tör20])**.** *The language of the generic limit set of any CA is Σ_3^0, and there exists a CA with a Σ_3^0-complete GLS, making that complexity bound tight.*

The aforementioned proposition notably gives a general bound of complexity on the language of any generic limit set, no matter its properties or the ones of its underlying cellular automaton.

Two additional recent results also provide insight on the complexity of GLS-related properties in general:

Proposition 2.3 (Th. 4.1 from [Del21])**.** *Any nontrivial property of the GLS of a CA is undecidable.*

Proposition 2.4 (Th. 8 from [Tör20])**.** *The decision problem of whether a GLS is a singleton, given as input the alphabet and local rule of its underlying CA, is* Σ_2^0*, and the bound is tight.*

These results can be compared to similar ones listed in [BDS10, Section 7] for omega-limit sets (based on works by Hurd and Kari) and μ-limit sets. Notably, the worst-case language for μ-limit sets is similarly Σ_3^0-complete, but the decision problem of being a singleton is easier in the GLS case. Moreover, in the case with equicontinuity points, the sets themselves are equal, due to the following remark.

Remark 10. As indirectly stated both in [BDP$^+$15, Prop. 4.1] and [DG19, Prop. 6.3], and even in a more restricted case as far as [KM00, Prop. 4], for any σ-ergodic measure μ with full support, for any CA with equicontinuity points, the μ-limit set and the generic limit set are equal. This is due to the fact that in that case, all words in the languages of both attractors come from configurations with infinitely many blocking words in both directions (a set that is comeager and of measure 1).

Notably, this is the case if μ is the usual Bernoulli probability measure on $\mathcal{A}^{\mathbb{Z}}$. As a consequence, the Σ_1^0-complete attractor on which f acts as the identity in Theorem 3 is both the GLS and the μ-limit set – and the bound from [BDP$^+$15, Th. 4.2] is reached.

In Section 3.2, we build – as mentioned in the previous remark – a cellular automaton to reach the upper complexity bound given by Proposition 2.11, for the worst case language of generic limit sets where f has equicontinuity points. However, we not only build it with that property, but with the more precise $f|_{\tilde{\omega}(f)} = \mathrm{id}|_{\tilde{\omega}(f)}$, which implies the existence of equicontinuity points. Consequently, we prove that the Σ_1^0-complete bound can be reached even with a more restrictive property.

Somewhat conversely, the upper complexity bound for generic limit sets where $f|_{\tilde{\omega}(f)}$ is a shift map is subtly obtained in Section 2.2.1. It is implied from a bound on a more general property, inclusion-minimality (Definition 2.5) which is broader than the previous assumption (see Corollary 3.15). That is to say, the complexities of even more GLSs share that Π_2^0 upper bound. That being said, the focus in the present subsection is still on the restriction "$f|_{\tilde{\omega}(f)}$ is a shift map", as it is a more studied property for its effects on omega-limit sets' and μ-limit sets' arithmetical complexity.

We end this section by mentioning how, in spite of recent works, Table 2.1 and its surrounding topic still have room for conjectures. For instance, the bound for μ-limit sets with a CA with equicontinuity points has yet to be reached. Most notably, a more detailed Rice-like theorem – determining the minimal complexity for nontrivial properties – on generic limit sets of cellular automata has yet to be obtained. Recent articles [Tör21, Del21] expect a behavior similar to the omega-limit and μ-limit sets, where the lowest complexity is reached with the problem of deciding whether a given attractor is a singleton. They believe it can be obtained through a careful construction adapted from [HdMS18], of which [Del21] has been the first meaningful step.

Conjecture 1. *Every nontrivial property of the generic limit set of a cellular automaton is* Σ_2^0*-hard.*

2.2 New complexity bounds

2.2.1 For inclusion-minimal automata

In this subsection, we define a specific attractor-related property, on which we then give a complexity bound.

Definition 2.5. An attractor A is *inclusion-minimal* if $\mathcal{D}(B)$ is meager for every closed set $B \subsetneq A$, or in other words, A does not properly contain another attractor.

To the current state of the art, we add Proposition 2.7, that requires a technical lemma before diving into its proof.

Lemma 2.6. *Let f be a CA on $\mathcal{A}^{\mathbb{Z}}$ with generic limit set $\widetilde{\omega}(f)$. The following conditions are equivalent:*

1. *$\widetilde{\omega}(f)$ is inclusion-minimal.*

2. *For all $s \in \mathcal{L}(\widetilde{\omega}(f))$, $v \in \mathcal{A}^*$ and $i \in \mathbb{Z}$, there are infinitely many $t \in \mathbb{N}$ with $f^t([v]_i) \cap [s] \neq \varnothing$.*

Note that if $s \notin \mathcal{L}(\widetilde{\omega}(f))$, then the v and i described in Item 2 cannot exist due to Lemma 1.16. The lemma characterizes the situation in which all choices of v and i are valid whenever one is. An equivalent formulation of Item 2 is that $\bigcup_{t \geq T} f^{-t}([s])$ is dense in $\mathcal{A}^{\mathbb{Z}}$ for all $s \in \mathcal{L}(\widetilde{\omega}(f))$ and $T \in \mathbb{N}$.

Proof. Suppose that Item 2 holds. For each word $s \in \mathcal{L}(\widetilde{\omega}(f))$ and $n \in \mathbb{N}$, the set $\bigcup_{t \geq T} f^{-t}([s]_n) = \sigma^{-n}(\bigcup_{t \geq T} f^{-t}([s]))$ is open, and dense by assumption. As a consequence, the intersection $B(s,n) = \bigcap_{T \in \mathbb{N}} \bigcup_{t \geq T} f^{-t}([s]_n)$ is comeager. Then $B = \bigcap_{s \in \mathcal{L}(\widetilde{\omega}(f))} \bigcap_{n \in \mathbb{N}} B(s,n) = \{x \in \mathcal{A}^{\mathbb{Z}} \mid \widetilde{\omega}(f) \subset \omega(x)\}$ is comeager as well, since any language is countable as a subset of $\mathcal{A}^*$. Consider any closed set $K \subset \mathcal{A}^{\mathbb{Z}}$. If there exists $x \in \mathcal{D}(K) \cap B$, then $\widetilde{\omega}(f) \subset \omega(x) \subset K$, so K is not a proper subset of $\widetilde{\omega}(f)$. Otherwise $\mathcal{D}(K) \subset \mathcal{A}^{\mathbb{Z}} \setminus B$ is meager. This means $\widetilde{\omega}(f)$ is inclusion-minimal.

Suppose then that Item 2 does not hold: there exist $s \in \mathcal{L}(\widetilde{\omega}(f))$, $v \in \mathcal{A}^*$, $i \in \mathbb{Z}$ and $T \in \mathbb{N}$ such that $f^t([v]_i)$ does not intersect $[s]$ for any $t \geq T$. Let $K = \widetilde{\omega}(f) \setminus [s]$, a closed proper subset of $\widetilde{\omega}(f)$. Then the basin $\mathcal{D}(K)$ contains $\mathcal{D}(\widetilde{\omega}(f)) \cap [v]_i$, which is nonmeager as the intersection of a comeager set and an open set. Hence $\widetilde{\omega}(f)$ is not inclusion-minimal. □

Now, for the actual proposition, based on the quantifiers used earlier:

Proposition 2.7. *Let f be a CA. If $\widetilde{\omega}(f)$ is inclusion-minimal, then its language is Π_2^0.*

Proof. Due to Item 2 of Lemma 2.6 and the small paragraph that follows, $\mathcal{L}(\widetilde{\omega}(f))$ is described by all s so that

$$\forall v \in \mathcal{A}^*, \forall i \in \mathbb{Z}, \forall T \in \mathbb{N}, \exists t > T, f^t([v]_i) \cap [s] \neq \varnothing.$$

Since f is computable, given all the parameters as input, checking whether $f^t([v]_i) \cap [s]$ is empty can be done with a Turing Machine. The full logical formula is therefore Π_2^0, and so is $\mathcal{L}(\widetilde{\omega}(f))$. □

2.2.2 For automata with equicontinuity points

The title of this subsection asks for a definition of the notion of equicontinuity points.

Definition 2.8. Let X be a compact metric space with metric d. Let (X, τ, f) be a dynamical system.

A point $x \in X$ is an *equicontinuity point* if

$$\forall \varepsilon > 0, \exists \delta > 0, \forall y \in X, d(x, y) < \delta \Rightarrow \forall n \in \mathbb{N}, d(f^n(x), f^n(y)) < \varepsilon.$$

(X, τ, f) is equicontinuous if all of its points are equicontinuity points.

Remark 11. Philosophically, an equicontinuity point x is so that any point in the dynamical system that starts close to x stays close to the trajectory $(f^n(x))_{n\in\mathbb{N}}$ as f is iterated.

Example 12. Consider $X = \{0, 1\}^{\mathbb{Z}}$ endowed with the prodiscrete topology (as used in Section 1.1.1). Add to it the following fitting distance:

$$d(x, y) := \inf_{n\in\mathbb{N}_0} \{2^{-n} \mid x([-n, n]) = y([-n, n])\}.$$

Let us use the cellular automaton f described in Example 1: the “spreading 1”.

Then $\ldots 111 \ldots$ is a fixpoint of the cellular automaton. It is also the sole element of the generic limit set $\widetilde{\omega}(f)$, and in that regard an obvious attractor of any configuration except $\ldots 000 \ldots$. As such, it is even more – an equicontinuity point. Considering its properties, it actually satisfies a strong version of the previous implication:

$$\forall \varepsilon > 0, \forall y \in X, d(\ldots 111 \ldots, y) < \varepsilon \Rightarrow d(\ldots 111 \ldots, f(y)) < \varepsilon.$$

Example 13. The automaton from Example 7 that spreads 1’s while shifting them to the left, however, has no equicontinuity point. Said briefly, it is because the distance d as in Example 12, being measured with respect to coordinate 0, does not fare well with the left shift operated by the function. Only the configurations $\ldots 000 \ldots$ and $\ldots 111 \ldots$ could be likely candidate for equicontinuity points, and it is easy to verify that they are not.

Definition 2.9. A word $b \in \mathcal{A}^{2k+1}$ is a *blocking word* for the CA f of radius $r \leq k$ if there exists a sequence of words $v_n \in \mathcal{A}^r$ such that for any $x \in [b]_{-k}$, we have $f^n(x) \in [v_n]$ for all $n \in \mathbb{N}$.

Remark 12. Since f is of radius $r \leq k$, a blocking word in a configuration completely disconnects coordinates to the right and left of the succession of v_n’s for the action of the automaton.

Remark 13. Note that b can be a blocking word without any v_n being one. For instance, a configuration containing v_1 with some specific surroundings that are not the ones obtained from $f(b)$ will not necessarily contain v_2 at the same position on the next step.

The next result is used in [Kůr97] and stated under its current form in [BT00, Prop. 2.1]:

Proposition 2.10. *A one-dimensional CA admits an equicontinuity point if and only if it admits a blocking word.*

Example 14. Based on the same cellular automaton as in Example 1, which was already said to have an equicontinuity point in Example 12, we can see that 111 is a blocking word for it.

The following result generalizes [Tör20, Prop. 2], where f was required to be equicontinuous when restricted to $\widetilde{\omega}(f)$.

Proposition 2.11. *Let f be a CA with equicontinuity points. The language of its generic limit set $\widetilde{\omega}(f)$ is Σ^0_1.*

Proof. A one-dimensional CA with equicontinuity points has at least one blocking word, by Proposition 2.10 [BT00]. Let $b \in \mathcal{A}^{2k+1}$ be a blocking word for f, and let $v_n \in \mathcal{A}^r$ for $n \in \mathbb{N}$ be the associated sequence of words, which is eventually periodic by the pigeonhole principle[1]: there are $N \geq 0$ and $p > 0$ with $v_{n+p} = v_n$ for all $n \geq N$. Let $i \in \mathbb{N}$ and consider a configuration $x \in [b]_{-i-k} \cap [b]_{i-k}$ – that is, which contains two carefully-spaced blocking words. We have $f^n(x) \in [v_n]_{-i} \cap [v_n]_i$ for all $n \in \mathbb{N}$, since each blocking word determines entirely the sequence that follows. As r is the radius of f, no information can "flow over" the v_n-words: since these are completely determined and larger than the radius r, no content on a cell on their left can influence anything on a cell on their right, and vice versa. Notably, the word $f^{n+1}(x)_{[-i+r,i-1]} \in \mathcal{A}^{2i-r+1}$, which is surrounded by two v_{n+1} in $f^{n+1}(x)$, is completely determined by $f^n(x)_{[-i,i+r-1]}$ for all n. Since the sequence $(v_n)_{n\in\mathbb{N}}$ is eventually p-periodic from index N onward, the sequence $\alpha_i(x) := (f^n(x)_{[-i,i+r-1]})_{n\in\mathbb{N}}$ is q-periodic from index $N + p|\mathcal{A}|^{2i-r+1}$ onward for some $q \leq p|\mathcal{A}|^{2i-r+1}$, and it only depends on $x_{[-i-k,i+k-1]}$.

Let $s \in \mathcal{A}^*$ be arbitrary. We claim that $s \in \mathcal{L}(\widetilde{\omega}(f))$ if and only if there exist $i \geq \max(|s|, r)$ and $N + p|\mathcal{A}|^{2i-r+1} \leq t \leq N + 2p|\mathcal{A}|^{2i-r+1}$ such that $f^{-t}([s]) \cap [b]_{-i-k} \cap [b]_{i-k} \neq \varnothing$. As this condition is Σ^0_1, the result follows.

Suppose first that the latter condition holds for some i and t, and let $x \in f^{-t}([s]) \cap [b]_{-i} \cap [b]_i$ be arbitrary. Denote $v = x_{[-i-k,i+k]}$, which begins and ends with b. We claim that v enables s in the sense of Lemma 1.16. For this, pick any $u, w \in \mathcal{A}^*$, and let $y \in [uvw]_{-i-k-|u|}$ be arbitrary. Since $y \in [b]_{-i} \cap [b]_i$, the sequence $\alpha_i(y) = \alpha_i(x)$ is periodic from index $N + p|\mathcal{A}|^{2i-r+1}$ onward. Hence $f^n(y)_{[0,|s|-1]} = f^n(x)_{[0,|s|-1]} = s$ holds for infinitely many n, and v enables s.

Conversely, suppose that the latter condition does not hold: for all $i \geq \max(|s|, r)$ and $N + p|\mathcal{A}|^{2i-r+1} \leq t \leq N + 2p|\mathcal{A}|^{2i-r+1}$ we have $f^{-t}([s]) \cap [b]_{-i} \cap [b]_i = \varnothing$. We show that no word $v \in \mathcal{A}^*$ enables s. Pick any $j \in \mathbb{Z}$ and let $i \in \mathbb{N}$ be so large that there exists $x \in [b]_{-i-k} \cap [v]_j \cap [b]_{i-k}$. By assumption $f^t(x)_{[0,|s|-1]} \neq s$ for all $N + p|\mathcal{A}|^{2i-r+1} \leq t \leq N + 2p|\mathcal{A}|^{2i-r+1}$. The sequence $\alpha_i(x)$ is q-periodic with $q \leq p|\mathcal{A}|^{2i-r+1}$ from index $N + p|\mathcal{A}|^{2i-r+1}$, so $f^t(x)_{[0,|s|-1]} \neq s$ holds for all $t \geq N + p|\mathcal{A}|^{2i-r+1}$. Hence v does not enable s. □

2.3 Results on shift-minimality

This section is dedicated to several additional results related to shift-minimality: notably, we study what this property tells us about generic limit sets, both in terms of arithmetical complexity (Corollary 2.16) and general structure (Proposition 2.15, and less directly Proposition 2.17 and Corollary 2.18).

Definition 2.12. Let X be a subshift. If X does not contain a nonempty proper subshift, we say that X is *shift-minimal*.

Remark 14. The most common terminology when studying subshifts is *minimal*; here, we keep a "shift-" prefix to ensure that no confusion arises with inclusion-minimality – a distinct, attractor-related notion described in Definition 2.5.

[1] Also known as "le principe pigeon-trou" in French.

Before stating new results, here are a few propositions based on transitivity and chain transitivity that are of use here (and in Section 3.4).

Proposition 2.13. *Let X be a subshift. If X is shift-minimal, then it is transitive.*

Proof. Suppose the subshift $X \subset \mathcal{A}^{\mathbb{Z}}$ is not transitive: there exist two words $u, v \in \mathcal{L}(X)$, so that no configuration in X contains both u and v as subwords. Consequently, for any big enough $n \in \mathbb{N}$, we have nested word sequences (u_n) and (v_n) so that $|u_n| = |v_n| = n$ and no configuration of X contains both; we can find them simply by extending u and v to bigger subwords – of the same length – of valid configurations. By compactness, we can extract configurations u and v of X that are not in each other's orbit closure, and so X is not shift-minimal. □

Proposition 2.14 (Prop. 6 from [Tör20])**.** *If a subshift is chain transitive and has a finite factor that does not consist of fixed points, then it is not the generic limit set of any CA.*

As a preamble to the following new results, it is vital to clearly understand the difference between shift-minimality (not properly containing a subshift, Definition 2.12) and inclusion-minimality (not properly containing an attractor of the CA, Definition 2.5): the former is a subshift-related property, while the latter is an attractor-related property.

In all generality, an attractor can be inclusion-minimal without even being a subshift, and thus without being shift-minimal. Even when both notions are defined, an attractor that is also a subshift can be inclusion-minimal without being shift-minimal, if it contains a subshift that is no attractor (for instance if said subshift attracts a meager portion of the space). Conversely, it can be shift-minimal without being inclusion-minimal, if it contains a specific sub-attractor that is not a subshift.

That being said, Proposition 2.15 shows that shift-minimality implies inclusion-minimality in a generic limit set. Conversely, Proposition 2.17 shows that any subshift contained in the generic limit set is necessarily *not* a topological attractor – meaning its basin of attraction is meager.

Proposition 2.15. *Let f be a CA on $\mathcal{A}^{\mathbb{Z}}$. If $\widetilde{\omega}(f)$ is shift-minimal, then it is inclusion-minimal.*

Proof. Suppose on the contrary that $X = \widetilde{\omega}(f)$ is not inclusion-minimal. Then it properly contains a closed set $K \subset X$ with a basin of attraction $B = \mathcal{D}(K) \subset \mathcal{A}^{\mathbb{Z}}$ that is not meager, and B is not comeager either (because $\widetilde{\omega}(f)$ is the generic limit set). Since K is closed, there exists $v \in \mathcal{L}(X)$ with $K \cap [v] = \varnothing$. As B has the Baire property, it is comeager in some nonempty open set, which we can choose to be a cylinder set $[w]_j \subset \mathcal{A}^{\mathbb{Z}}$ where w is not the empty word. Our goal is to show that v occurs periodically in every configuration of X, use these occurrences to construct a factor map onto a finite dynamical system, and obtain a contradiction with Proposition 2.14.

Denote $p = |w| > 0$. For each $n \geq 0$, the basin of $K_n = \bigcap_{i=-n}^{n} \sigma^{ip}(K)$, which equals $\bigcap_{i=-n}^{n} \sigma^{ip}(B)$, is comeager in the cylinder set $[w^{2n+1}]_{j-np}$. In particular, each K_n is nonempty, hence their intersection $K' = \bigcap_{i\in\mathbb{Z}} \sigma^{ip}(K) \subset X$ is nonempty as well. Since K is disjoint from $[v]$, we have $K' \subset \bigcap_{i\in\mathbb{Z}} (\mathcal{A}^{\mathbb{Z}} \setminus [v]_{ip})$. For $P \subset \mathbb{Z}$, define $X(P) = X \cap \bigcap_{i\in P} (\mathcal{A}^{\mathbb{Z}} \setminus [v]_i)$. We saw that $X(P) \neq \varnothing$ for some infinite subgroup $P = p\mathbb{Z} \subset \mathbb{Z}$. If we had $p = 1$, $X(P)$ would be closed and stable by σ, hence a subshift contained in X. By shift-minimality of X, this means $X(P) = X$, which contradicts the nonemptiness of $[v] \cap X$. Thus $p \geq 2$.

Let $q \geq 2$ be minimal such that $X(q\mathbb{Z}) \neq \varnothing$: there are configurations of X with no subword v starting on indices in $q\mathbb{Z}$. For $x \in X$, let $C(x) = \{q\mathbb{Z} + i \mid x \in X(q\mathbb{Z} + i)\}$ be the set of cosets on which x does not contain occurrences of v. Then $C(\sigma(x)) = C(x) + 1 = \{q\mathbb{Z} + i + 1 \mid q\mathbb{Z} + i \in C(x)\}$

for all x. The number $|C(x)|$ is the same for all $x \in X$: the set X' of those configurations x for which $|C(x)|$ is maximal forms a subshift of X, and by shift-minimality of X we have $X' = X$. Denote $m = |C(x)|$.

The sets $C(x), C(\sigma(x)), \ldots, C(\sigma^{q-1}(x))$ are distinct for all $x \in X$: if $C(x) + a = C(\sigma^a(x)) = C(\sigma^b(x)) = C(x) + b$ for some $0 \leq a < b < q$, then $C(x) + (b-a) = C(x)$, meaning that $\sigma^{-a}(x) \in X((b-a)\mathbb{Z})$, contradicting the minimality of q. Also, there exists $r \geq 0$ such that $C(x)$ only depends on $x_{[-r,r]}$: otherwise for all $r \geq 0$ there would exist $x(r) \in X \cap \bigcap_{i \in I} X(q[-r,r]+i)$ with $I \subset [0, q-1]$ of cardinality at least $m+1$, and a limit point x of $(x(r))_{r \geq 0}$ would satisfy $C(x) \geq m+1$, a contradiction. All in all, we have shown that $C \colon (X, \sigma) \to (2^{\{0,\ldots,q-1\}}, +1)$ is a morphism of dynamical systems whose image is finite and contains no fixed points. Since X is shift-minimal, it is transitive (see Proposition 2.13) and notably chain transitive; this contradicts Proposition 2.14. □

Together with Proposition 2.15, Proposition 2.7 implies that a shift-minimal generic limit set has a Π^0_2 language. In [Tör20, Prop. 1] it was proved to be Σ^0_2, hence it must be Δ^0_2.

Corollary 2.16. *Let f be a CA. If $\widetilde{\omega}(f)$ is shift-minimal, then its language is Δ^0_2.*

Now, we show that even though the generic limit set of a CA might properly contain closed sets with nonmeager basins, these sets cannot be subshifts. In fact, we can characterize the generic limit set as the smallest subshift with a nonmeager basin.

Proposition 2.17. *Let f be a CA on $\mathcal{A}^{\mathbb{Z}}$ and $X \subset \mathcal{A}^{\mathbb{Z}}$ a subshift. If $\mathcal{D}(X)$ is nonmeager, then $\widetilde{\omega}(f) \subset X$ (and $\mathcal{D}(X)$ is consequently comeager).*

Proof. Suppose for a contradiction that $\widetilde{\omega}(f) \setminus X \neq \varnothing$. Then there exists a word $s \in \mathcal{L}(\widetilde{\omega}(f)) \setminus \mathcal{L}(X)$. Let $v \in \mathcal{A}^*$ and $i \in \mathbb{Z}$ be given by Lemma 1.16 applied to s. Similarly to the proof of Lemma 2.6, the set $B(s) = \bigcap_{T \in \mathbb{N}} \bigcup_{t \geq T} f^{-t}([s])$ is comeager in $[v]_i$.

As X is closed, its basin $\mathcal{D}(X)$ has the Baire property, and since $\mathcal{D}(X)$ is by assumption nonmeager, it is comeager in some nonempty open set $U \subset \mathcal{A}^{\mathbb{Z}}$. Moreover, since the shift map σ is a homeomorphism and commutes with f, for any $x \in \mathcal{D}(X)$, $\sigma(x)$ is so that $\omega(\sigma(x)) = \sigma(\omega(x)) \subset X$. As such, $\mathcal{D}(X)$ is stable by σ. We can assume that $V = [v]_i \cap U \neq \varnothing$ – at worst, rename as U some $\sigma^k(U), k \in \mathbb{Z}$ in which $\mathcal{D}(X)$ is also comeager. Hence $\mathcal{D}(X) \cap B(s)$ is comeager in V, in particular nonempty. Any configuration x in this set satisfies $\omega(x) \subset X$ and $\omega(x) \cap [s] \neq \varnothing$, so X intersects $[s]$. This contradicts $s \notin \mathcal{L}(X)$. □

From the previous two propositions, we deduce the following:

Corollary 2.18. *Let f be a CA on $\mathcal{A}^{\mathbb{Z}}$. If $\widetilde{\omega}(f)$ contains a topological subattractor, then it contains a nonempty proper subshift, that necessarily has a meager basin of attraction.*

Proof. If $\widetilde{\omega}(f)$ contains a topological subattractor, then it is not inclusion-minimal. Hence by Proposition 2.15, it is not shift-minimal, and therefore it contains a nonempty proper subshift X. But by Proposition 2.17 $\mathcal{D}(X)$ must be meager, else it wouldn't be proper. □

Chapter 3

Realization of Subshifts

This chapter, based on the second half of [ENT22], realizes the bounds mentioned in Section 2.2. It does so through three sections of increasing sophistication.

First, Section 3.1 presents a broad method for building complex generic limit sets. The basic idea is simple: it aims to implement a Turing Machine in the cellular automaton that would approach a set with the desired arithmetical complexity. However, we need a symbol that would indicate where the head starts, and restricting the starting configurations to the ones containing exactly one such symbol is in total contradiction with the "generic" aspect of the generic limit set, where most configurations must converge toward such a complex attractor. Therefore, the biinfinite tape of the cellular automaton is divided into finite segments, that are portions between two initializing seeds $\mathcal{I}$. Each of these seeds cleans the segment to its left then implements a Turing Machine where it stands, with the segment to its left as a (space-limited) tape. This is what we call the generic construction.

Second, Section 3.2 proves that the bound of Proposition 2.11 is tight, by constructing, based on the generic construction, a generic limit set of complexity Σ_1^0 that has an equicontinuity point. Basically, each segment is filled with an appropriate Turing Machine that repeats the same "complicated enough" patterns, and keeps to itself.

Third, Section 3.3 realizes two large classes of Π_2^0 subshifts generic limit sets: the ones containing a Π_1^0 subshift, and the ones of complexity Δ_2^0. That result requires its fair share of subsections to be proven, as it uses a refined merging process between segments from the generic construction. Nonetheless, it proves that the bound from Proposition 2.7 is tight. Though it leaves open whether any Π_2^0 subshift can be realized as a GLS, it has its interesting lot of corollaries, as detailed in Section 3.4.

3.1 Generic construction

In this section we present a construction of a CA f which serves as a base for the CA built in Section 3.2 and Section 3.3, where within each proof modifications are introduced. This type of construction first appeared in [DPST11]; our presentation is based on [BDP+15]. An even more complex version was presented in [HdMS18].

The main idea is the following: the alphabet $\mathcal{A}$ of f is the cartesian product of several auxiliary alphabets regarded as layers. The biinfinite tape, using these layers, is divided into individual finite computation zones called segments where the computations occur after the deletion of most of the initial data. The computations depend on the application at hand: we simulate Turing Machines in Section 3.2 and store patterns from subshifts in Section 3.3. Depending on the construction, some segments may be merged with other segments as time passes.

The aforementioned layers of $\mathcal{A}$ are:

- *Main Layer* $\mathcal{A}_{\text{main}}$. It is the layer that contains the alphabet of the subshift we want to obtain as GLS – the one on which what we want to simulate is displayed, hence the main one. Three special additional symbols are also included in it: *walls symbols* $\mathcal{W}$, *initialization symbols* $\mathcal{I}$, and *blank symbols* \$. An initialization $\mathcal{I}$-symbol is turned into a $\mathcal{W}$-symbol at the first step of the automaton, and two successive $\mathcal{W}$-symbols delimit areas of computation called segments. As time goes by, desired patterns are written on this layer as needed.

- *Computation Layer* $\mathcal{A}_{\text{comp}}$. It encodes, in each segment as delimited on the Main Layer, a Turing Machine $\mathcal{M}$ which carries over the desired computations, and possibly other tasks. The simulated $\mathcal{M}$ writes the results of its computation on the Main Layer (the details vary depending on the application).

- *Cleaning Layer* $\mathcal{A}_{\text{clean}}$. Using several types of signals, this layer erases any relic from the initial configuration.

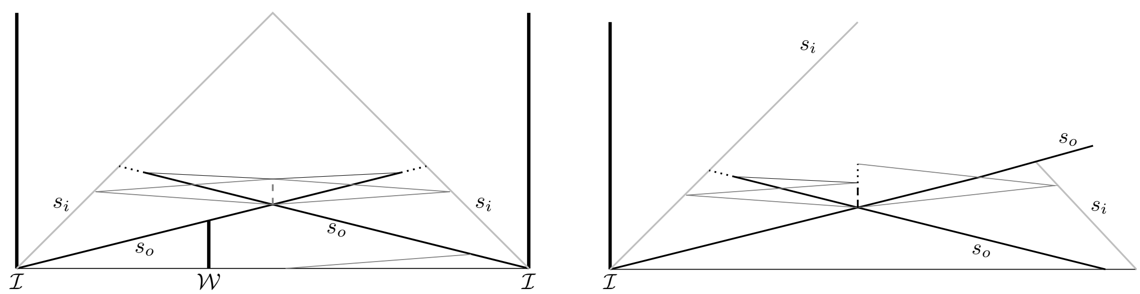
In each alphabet we have a blank symbol which replaces data that is said to be 'erased' – for instance, in the Main Layer this role is played by \$. We denote by π_{main}, π_{comp}, and π_{clean} the projections on the Main, Computation, and Cleaning Layers, respectively. We also formally define the following:

Definition 3.1. Let $x \in \mathcal{A}^{\mathbb{Z}}$ be a configuration and consider the forward orbit $(f^n(x))_{n\geq 0}$. A *segment* in the initial configuration x is a sequence of successive cells $s(i,j) = x_i x_{i+1} \dots x_{j-1} x_j$ such that $\pi_{main}(x_i) = \pi_{main}(x_j) = \mathcal{I}$ and $\pi_{main}(x_k) \neq \mathcal{I}$ for all $i < k < j$. For $n \geq 1$, a segment in $y = f^n(x)$ is a sequence $y_i y_{i+1} \dots y_{j-1} y_j$ such that $y_i = y_j = \mathcal{W}$ and $s(i,j)$ is a segment of x.

In order for all segments to start and perform their computation without disruption, it is necessary to clean the data on all layers in the initial configuration, with the exception of $\mathcal{I}$-symbols in $\mathcal{A}_{\text{main}}$ – which initiate the cleaning and the segments' internal processes, and are immediately turned into $\mathcal{W}$-symbols at the first step of the CA. Observe that walls may also be present in the initial configuration, *i.e.*, some $\mathcal{W}$-symbols are not created by an $\mathcal{I}$-symbol. These walls need to be deleted.

The deletion process is carried out by two signals s_i and s_o (inner and outer) generated by every initialization symbol $\mathcal{I}$ in both directions, on the Cleaning Layer $\mathcal{A}_{\text{clean}}$. They are erased once they meet their counterpart coming from another $\mathcal{I}$-symbol, and they delete any wall and data they encounter. These signals are defined similarly to [BDP+15, Section 3], which ensures that correct moving speed for these kinds of signals exist. The rules they obey are the following: the outer signal s_o travels faster than s_i; s_o deletes everything it encounters on each layer that is not another s_o; when two s_o signals collide, they send auxiliary signals (thin gray signals in Fig. 3.1) that bounce back on the inner signals s_i behind them and return to the collision point. If the bouncing signals do not meet again at the same time step, as pictured in Fig. 3.1b, the side from which the latter one came has a greater gap between its two signals s_o and s_i, meaning this pair has not been generated at time 1 – this holds since signals s_o and s_i can not be both present in the same cell, and since their gap grows with each time step. The pair with the greater gap is deleted by implementing a simple priority rule of s_o over s_i. The pair (s_o, s_i) with the smaller gap is preserved by the use of some auxiliary signal (thin black signal in Fig. 3.1b) generated by the first-arrived bouncing signal: that auxiliary signal weakens the antagonistic s_o signal that could have disrupted the s_i signal from the pair. If the bouncing signals do meet again at the same time step, as pictured in Fig. 3.1a, they are both deleted. Auxiliary signals (thin black signals in Fig. 3.1a) delete both s_o's, and both s_i's cancel each other when meeting. As a result, since this situation notably happens when signals coming from two $\mathcal{I}$ symbols meet midway, the entire segment between the walls these $\mathcal{I}$ generate has been cleaned from any possible disruption, and the Turing Machine initialized on the Computation Layer can run without a hitch.

Consequently, our construction – inspired by [BDP+15] protects walls originating from $\mathcal{I}$-states, and deletes any other wall. Note that the signals s_o and s_i need to move slower than speed 1 (one cell at each time step) for the process with bouncing signals to go smoothly: speeds 1/4 and 1/5 work according to [BDP+15]. This requires several states for the signals, to indicate each "movement" of length 1/4 or 1/5.



(a) Pairs of signals s_o and s_i from two adjacent $\mathcal{I}$'s meet. The s_o's erase everything else of the original configuration in the segment (here, a starting $\mathcal{W}$ and a lonely auxiliary signal).

(b) "Wild" signals from the original configuration cannot disrupt a pair of s_o and s_i coming from an $\mathcal{I}$. The slope of some auxiliary signals is slightly exaggerated for the phenomenon of them bouncing back not at the same time to be more visible.

Figure 3.1: Space-time diagram of the deleting process.

At time step 1, as it is turned into a $\mathcal{W}$-symbol and launches signals s_i and s_o, each $\mathcal{I}$-symbol also starts an internal computation process on $\mathcal{A}_{\text{comp}}$ in its associated segment. These internal computation processes vary for each construction, but in any case they have a clean canvas to perform any computation needed, as the outer signals s_o will replace the contents of each correctly initialized segment with blank symbols.

3.2 Realization of complexity for equicontinuity points

In this section, we build a CA f which realizes the bound in Proposition 2.11, that is, f has equicontinuity points and the language of $\widetilde{\omega}(f)$ is Σ_1^0-complete. Moreover, we show that such an f can be built so that it acts as the identity on its generic limit set.

In the following result, note that $[\mathcal{L}_n(\widetilde{\omega}(f))]$ is the set of configurations $x \in \mathcal{A}^{\mathbb{Z}}$ with $x_{[0,n-1]} \in \mathcal{L}(\widetilde{\omega}(f))$.

Lemma 3.2 (Lemma 3 of [Tör20]). *Let f be a CA on $\mathcal{A}^{\mathbb{Z}}$, let $n \in \mathbb{N}$ and let $[v]_i \subset \mathcal{A}^{\mathbb{Z}}$ be a cylinder set. Then there exists a cylinder set $[w]_j \subset [v]_i$ and $T \in \mathbb{N}$ such that for all $t \geq T$ we have $f^t([w]_j) \subset [\mathcal{L}_n(\widetilde{\omega}(f))]$.*

We say that w is $\widetilde{\omega}(f)$-forcing.

Lemma 3.3. *Let f be a CA so that $f|_{\widetilde{\omega}(f)} = \mathrm{id}|_{\widetilde{\omega}(f)}$. Then f has an equicontinuity point on $\mathcal{A}^{\mathbb{Z}}$.*

Proof. Choose a $\widetilde{\omega}(f)$-forcing word $w \in \mathcal{A}^*$ with $f^t([w]_j) \subset [\mathcal{L}_{3r}(\widetilde{\omega}(f))]$ for all $t \geq T$. Then every $x \in [w]_{j-r}$ satisfies $f^{t+1}(x)_{[0,r-1]} = f^t(x)_{[0,r-1]}$ for all $t \geq T$. Thus we can extend w into a blocking word and thus f has an equicontinuity point on $\mathcal{A}^{\mathbb{Z}}$. □

Theorem 3. *There exists a CA f such that $\widetilde{\omega}(f)$ is a Σ_1^0-complete set and $f|_{\widetilde{\omega}(f)} = \mathrm{id}|_{\widetilde{\omega}(f)}$.*

Proof. We describe a CA f with the desired properties. Consider the construction from Section 3.1, modified as follows.

- The Main Layer's alphabet is $\{0, 1, \$, \mathcal{I}, \mathcal{W}\}$, where \$ is the blank symbol.
- The only way to erase a wall is with an outer signal s_o. In particular, walls created by an $\mathcal{I}$ always remain, so that segments formed between two of them stay forever.
- In addition to the signals s_o and s_i, each $\mathcal{I}$ initializes a simulated computation of a Turing Machine $\mathcal{M}$ on the segment to its left.
- The Cleaning Layer and its deleting process, described in Section 3.1, remain untouched.
- As the cleaning and deleting processes take place, all the information in any segment is replaced by \$-symbols.

As for the behavior of the machine $\mathcal{M}$, consider an enumeration of all Turing Machines $(\mathcal{M}_n)_{n\in\mathbb{N}}$ with a one-way infinite tape, and consider a computable bijection $p : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ – for instance, the inverse of the Cantor pairing function, but we modify it so that we avoid any case where $\ell - (|\mathrm{bin}(n)| + 1) < 0$ with $(n, m) = p(\ell)$, where $\mathrm{bin}(n)$ is the binary representation of n.

In each segment, $\mathcal{M}$ starts by determining the length ℓ of its segment (by sending a specific signal and waiting for its return, for instance) and computes $(n, m) = p(\ell)$. Then $\mathcal{M}$ simulates m steps of computation of the machine $\mathcal{M}_n$ on the empty input. If $\mathcal{M}_n$ halts during these m simulated steps, then $\mathcal{M}$ prints $\mathrm{bin}(n) \in \{0, 1\}^*$ on the left end of the segment, leaving one blank cell between it and the left wall, and fills the rest of the segment with blank symbols. If $\mathcal{M}_n$ does not halt in at most m steps of computation, $\mathcal{M}$ fills the segment with blank symbols. In both cases, once the described computations are done, $\mathcal{M}$ is deleted. In this manner, every segment is eventually of the form $\mathcal{W}\$\mathrm{bin}(n)\$^k\mathcal{W}$ or $\mathcal{W}\$^\ell\mathcal{W}$, with $k = \ell - (|\mathrm{bin}(n)| + 1)$ (notice that p is designed so that $k \geq 0$). The segment remains unchanged from that point on.

We first claim that $\widetilde{\omega}(f) \subset (\{\mathcal{W}, 0, 1, \$\} \times \{\$\} \times \{\$\})^{\mathbb{Z}}$. Once proved, this implies $f|_{\widetilde{\omega}(f)} = \mathrm{id}|_{\widetilde{\omega}(f)}$, since f acts as the identity on the above full shift. Let $s \in \mathcal{L}(\widetilde{\omega}(f))$ be enabled by some cylinder set $[v]_i$ as per Lemma 1.16. We may assume, by extending v if necessary, that $i \leq 0$ and $|v| \geq |i| + |s|$. Choose $u = w = \mathcal{I}$. Then for any configuration $x \in [uvw]_{i-1}$, the Cleaning Layer ensures that the word $\pi_{\mathrm{main}}(f^t(x)_{[i-1,i+|v|+1]}) \in \mathcal{W}\mathcal{A}^{|v|}\mathcal{W}$ consists of correctly initialized segments for all large enough $t \geq 1$. The instances of the machine $\mathcal{M}$ simulated on the Computation Layer will eventually fill each segment with symbols from $\{0, 1, \$\}$ and disappear. Thus $f^t([uvw]_{i-1}) \subset [\mathcal{W}\{\mathcal{W}, 0, 1, \$\}^{|v|}\mathcal{W}]_{i-1}$

for all large enough t, and infinitely many of them contain $[s]$ due to v enabling s, and thus $s \in \{\mathcal{W}, 0, 1, \$\}^*$.

Let $n \in \mathbb{N}$. We claim that $s_n \in \mathcal{L}(\widetilde{\omega}(f))$, with $\pi_{\text{main}}(s_n) = \$\text{bin}(n)\$$ if and only if $\mathcal{M}_n$ eventually halts. First, if $\mathcal{M}_n$ never halts, then no correctly initialized segment will contain the word s_n. By the analysis in the previous paragraph and the construction above, $s_n \notin \mathcal{L}(\widetilde{\omega}(f))$. Suppose now that $\mathcal{M}_n$ halts in some m steps. Since p is a bijection from $\mathbb{N}$ to $\mathbb{N} \times \mathbb{N}$, there exists ℓ such that $(n, m) = p(\ell)$. We show that s_n is enabled by any cylinder C such that $\pi_{\text{main}}(C) = [\mathcal{I}\$^\ell\mathcal{I}]_{-1}$. For all $x \in C$ and $t \geq 1$, the word $\pi_{\text{main}}(f^t(x)_{[-1,\ell]}) \in \mathcal{W}\mathcal{A}^\ell\mathcal{W}$ is a correctly initialized segment; and the instance of $\mathcal{M}$ it contains simulates m steps of $\mathcal{M}_n$ on that segment. When $\mathcal{M}_n$ halts, $\mathcal{M}$ writes $\$\text{bin}(n)\$^{\ell-(|\text{bin}(n)|+1)}$ on the segment and disappears. Hence $\pi_{\text{main}}(f^t(x)) \in [\$\text{bin}(n)\$]$ for all large enough t.

Since the set of Turing Machines that eventually halt on the empty input is known to be Σ_0^1-complete, we have built the expected CA. $\square$

Corollary 3.4. *There exists a CA f such that $\widetilde{\omega}(f)$ is a Σ_1^0-complete set and f has an equicontinuity point.*

3.3 Realization of a large class of Π_2^0 subshifts

3.3.1 Statement and overview of the proof

In this section, we realize two large classes of Π_2^0 subshifts as generic limit sets of cellular automata. More specifically, we prove the following result:

Theorem 4. *Let $X \subset \mathcal{A}^\mathbb{Z}$ be a chain mixing subshift satisfying one of the following conditions:*

1. *either X is Π_2^0 and contains a nonempty Π_1^0 subshift;*
2. *or X is Δ_2^0.*

Then there exists an alphabet $\mathcal{B} \supset \mathcal{A}$ and a CA $f : \mathcal{B}^\mathbb{Z} \to \mathcal{B}^\mathbb{Z}$ with $\widetilde{\omega}(f) = X$ and $f|_X = \sigma|_X$.

The two cases of the theorem are proven mostly simultaneously; we point out the (relatively minor) differences in the construction and proofs whenever they diverge. We call them the $\Pi_1^0 \subset \Pi_2^0$ case and the Δ_2^0 case.

Though in a more complex fashion than Section 3.2, what we prove here is also based on the generic construction of Section 3.1: several layers are used to build the cellular automaton, notably a Cleaning Layer to erase unwanted data from the starting configuration, a Computation Layer to obtain better and better approximations of configurations from the Π_2^0 (resp. Δ_2^0) subshift we want to realize, and the Main Layer that bears these resulting words. Just as it is broadly described in Section 3.1, most initial configurations here contain on their Main Layer an infinite number (on both directions) of cells with a symbol $\mathcal{I}$ that initialize a segment. Each of these segments cleans its initial cells, then runs a Turing Machine that prints words form the language of SFTs that are approximations of the desired Π_2^0 (resp. Δ_2^0) subshift.

The first two lemmas that follow, Lemma 3.5 and Lemma 3.6, show that we can approximate the language of a Π_2^0 (resp. Δ_2^0) subshift X using a carefully chosen sequence of mixing SFTs $(X_m)_{m\in\mathbb{N}}$, and words $(w_m)_{m\in\mathbb{N}}$ from their language that somehow contain any subword needed to describe the language of X asymptotically. The idea is that the Turing Machine in each segment will compute the best approximation X_m of X it can given the space it has, then print a word u whose periodic repetition is in X_m, and that contains the subword w_m – such a periodic configuration exists due to Lemma 3.7, and its least period n is ensured to be bigger as m grows. Once it has obtained such a u, the Turing Machine keeps printing it periodically, and simultaneously starts capturing – using somewhat "porous" walls – what it receives from the segment to its right, which after some time also periodically prints a word u' containing some $w_{m'}$. The synchronicity of these processes is ensured by a sub-layer of the Computation Layer, the Clock Layer, that runs a ternary counter; and by Lemma 3.10.

Looking at truncates of the word u' and using Lemma 3.8, the receiving Turing Machine can compute after a somewhat short time whether the least period of u' is smaller than the one of u. If that is the case, due to their dependence with the size of the segment and the index of the SFT approximations, it deduces that the segments to its right is smaller – and therefore has computed a *less precise* approximation of X than its own – and in that case keeps repeating that probing process without doing anything more. However, if the Turing Machine receives a word u' with a bigger period than its own u, that means the present segment has to merge with the one to its right, as the latter computed a better approximation of X. To do so, as described in Section 3.3.6, it erases its right wall and lets itself be overwritten by the segment on its right. That process happens for any segment eventually, and goes well, see Lemma 3.11, Lemma 3.12 and Lemma 3.13.

The remaining challenge is, first, to implement this rewriting, and second, to do so in such a way that it does not produce additional words in the generic limit set. Therefore, we need to glue together the words u and u' with care. The control we have over the mixing distances of X_m and $X_{m'}$ ensures that said process can be done within a relatively short time span – a property we need since all words we manipulate keep shifting to the left, and we consequently don't want them to drift away before the end of that computation. Notice how the chain mixing hypothesis is of use here, since it is at the origin of X_m and $X_{m'}$ being mixing, and therefore allows this capacity of gluing words in the first place. Besides, contrary to a mere property of transitivity, it allows that gluing to be done with any distance big enough between the two words. As a side note here, the chain mixing property turns out to be in fact necessary in the present construction, by Proposition 3.18.

The remaining problem is to glue u and u' together while not adding words to the generic limit set, since the latter must have the language of X. In the Δ_2^0 case, the approximations X_m are SFT approximations $\mathcal{S}_k(X)$, and thus are nested – each approximation is finer than the previous one, and $\mathcal{L}(X) = \lim_n \mathcal{L}(X_m)$. Therefore, gluing u and u' using a word from the least fine approximation is enough for the whole glued pattern to belong to some X_m, and asymptotically with bigger and bigger segments merging we get a finer and finer approximation of X, that is the only subshift whose language remains in the generic limit set.

However, in the $\Pi_1^0 \subset \Pi_2^0$ case, the X_m's are not nested into each other, and may have words they do not have in common, since the language of X is only the limsup of theirs. Therefore, gluing them is not as immediate as in the previous case! To circumvent that, we build additional subshifts of each X_m, the Y_m's, that are nested approximations of Y – this is why we need the inclusion of a Π_1^0 subshift Y in X to begin with. When the merging happens, as described in Section 3.3.6, we check one additional property of u': that it belongs to $\mathcal{L}(Y_m)$. Then, the gluing process glues u and u' as words of some Y_ℓ, using a word from that same SFT; and so with bigger and bigger segments the gluing words are closer and closer approximations of words of $Y \subset X$, resulting in a generic limit set that, as a whole, does describe the language of X.

3.3.2 Necessary lemmas

We begin with two technical lemmas that provide well-behaved SFT approximations of the subshifts that appear in the statement of Theorem 4.

Lemma 3.5. *Let $X \subset \mathcal{A}^{\mathbb{Z}}$ be a nonempty chain mixing Π_2^0 subshift, and $Y \subset X$ a nonempty Π_1^0 subshift. Then there exists a sequence $(X_m, Y_m, w_m)_{m \in \mathbb{N}}$, where each $X_m \subset \mathcal{A}^{\mathbb{Z}}$ is a mixing SFT, $Y_m \subset X_m$ is a nonempty SFT, and $w_m \in \mathcal{L}(X_m)$, such that the following conditions hold.*

1. *$Y = \bigcap_{m \in \mathbb{N}} Y_m$ and $Y_{m+1} \subset Y_m$ for all $m \in \mathbb{N}$.*
2. *$\mathcal{L}(X) = \bigcap_{M \in \mathbb{N}} \bigcup_{m \geq M} \mathcal{L}(X_m) = \bigcap_{M \in \mathbb{N}} \bigcup_{m \geq M} \{w_m\}$.*
3. *For each m, the window size and mixing distance of X_m, the window size of Y_m, and the length $|w_m|$ are all $o(\log m)$.*
4. *The function $m \mapsto (X_m, Y_m, w_m)$ is computable in $O(2^m)$ space.*

Proof. Since X is Π_2^0, there exists a computable predicate φ_X such that $\mathcal{L}(X) = \{w \in \mathcal{A}^* \mid \forall k \; \exists \ell \; \varphi_X(w, k, \ell)\}$. Since Y is Π_1^0, there exists a computable predicate φ_Y such that $\mathcal{L}(Y) = \{w \in \mathcal{A}^* \mid \forall k \; \varphi_Y(w, k)\}$. We first describe an algorithm that produces a sequence of SFTs satisfying the first two items. Then we modify it to satisfy the remaining items as well.

The algorithm keeps track of three finite sets of words $M, Q, F \subset \mathcal{A}^*$, which we call the memory, the queue, and the forbidden set. All three sets are initially empty. The memory and queue are used to construct the X_m and w_m, while the forbidden set is used for Y_m. For each $w \in M$, the algorithm also stores numbers $k_w, \ell_w \in \mathbb{N}_0$, and for each $w \in Q$ it stores a number $k'_w \in \mathbb{N}_0$.

The algorithm proceeds in rounds, starting from $i = 0$. Round i consists of the following steps:

1. Add a new word $u \in \mathcal{A}^* \setminus M$ to M; they are added in increasing order of length, and lexicographically for a given length. Set $k_u = k'_u = \ell_u = 0$.
2. For each $w \in M$, check whether $\varphi_X(w, k_w, \ell_w)$ holds. If it does, we say that w fires, and we increment k_w and set $\ell_w = 0$. If w is not an element of Q, we also add it to Q and set $k'_w = k_w$. If $\varphi_X(w, k_w, \ell_w)$ does not hold, we increment ℓ_w.
3. For each $w \in Q$, do the following. Let Y' be the SFT defined by forbidding all words in the forbidden set F. Denote $p = \max(k'_w, |w|)$ and $F_p = \{v \in \mathcal{A}^{\leq p} \mid k_v \leq p\}$. If the SFT defined by forbidding F_p contains a mixing sub-SFT X' with $w \in \mathcal{L}(X')$ and $Y' \subset X'$, then remove w from Q and output the triple (X', Y', w).
4. For each $w \in \bigcup_{j \leq i} \mathcal{A}^j$, if there exists $k \leq i$ such that $\varphi_Y(w, k)$ does not hold, then add w to F.

The algorithm executes these rounds in an infinite loop. It outputs a sequence of triples, which we denote by $(X_m, Y_m, w_m)_{m \in \mathbb{N}}$.

We explain why the existence of X' on Step 3 is algorithmically decidable. Let $Z \subset \mathcal{A}^{\mathbb{Z}}$ be the SFT defined by forbidding F_p. It has a finite number of maximal transitive subshifts, called its *transitive components*, which are likewise SFTs [LM95, Section 4.4]. The transitive components that happen to be mixing are the maximal mixing subshifts of Z. Hence, if X' exists, we can choose it among these finitely many components. The components and their relevant properties are all computable from F_p (see Sections 3.4, 4.4 and 4.5 of [LM95]).

By construction, each X_m produced by the algorithm is a mixing SFT with $w_m \in \mathcal{L}(X_m)$ and $Y_m \subset X_m$. Since the algorithm never removes words from F, the sequence $(Y_m)_{m\in\mathbb{N}}$ is decreasing. Step 4 of each round guarantees that every $w \in \mathcal{A}^* \setminus \mathcal{L}(Y)$ is eventually added to F, so Item 1 of the statement holds.

Consider then a word $w \in \mathcal{L}(X)$. Due to the definition of φ_X, it fires an infinite number of times during the execution of the algorithm. Whenever w fires and is not in the queue, it is added there and the number k'_w is fixed for all rounds until w leaves the queue. Consider then $p = \max(|w|, k'_w)$ and the set F_p; observe that while w is in the queue, p is fixed and F_p can only reduce in size. We prove that w does leave the queue after some round.

Because X is chain mixing, the SFT approximation $\mathcal{S}_p(X)$ is mixing, and its language contains w since $p \geq |w|$. If w did not leave the queue before that due to some output (X_m, Y_m, w_m) with $w_m = w$, each word u in $\mathcal{L}_{\leq p}(X)$ will eventually leave F_p (because its k_u grows to infinity with the rounds), and each word in $\mathcal{A}^{\leq p} \setminus \mathcal{L}(Y)$ will eventually enter F (by definition of Step 4). Once this happens, we have $Y' \subset \mathcal{S}_p(X)$. Thus we can choose $\mathcal{S}_p(X)$ as X' if a suitable mixing SFT was not found earlier, and the algorithm outputs (X', Y', w) as (X_m, Y_m, w_m) for some m. Therefore w is removed from the queue after some round.

Furthermore, any such w is added again at a later round when it eventually fires anew, since $w \in \mathcal{L}(X)$. Thus $w \in \bigcap_{M\in\mathbb{N}} \bigcup_{m\geq M} \{w_m\}$. In particular the algorithm produces an infinite sequence of triples.

Now, take a word $w \notin \mathcal{L}(X)$, which fires only a finite number of times. Denote $n = |w|$. After some number of rounds, each word $u \in \mathcal{A}^{\leq n} \setminus \mathcal{L}(X)$ has fired for the last time and the values k_u and k'_u have settled into constants. These words may leave the queue once more, but produce a finite number of outputs (to which w may belong) by doing so.

Let $K_n = \max\{k_u \mid u \in \mathcal{A}^{\leq n} \setminus \mathcal{L}(X)\}$. After a bigger number of rounds, whenever a new word $v \in \mathcal{A}^*$ fires and enters the queue, we have either $|v| > K_n$ or $k'_v > K_n$. Indeed, after some point, we have that $k_v > K_n$ holds for all $v \in \mathcal{L}_{\leq K_n}(X)$. This means that w will never leave the set F_p for $p = \max(|v|, k'_v)$, and thus does not occur in the mixing SFT X' if one is produced for such a v.

Hence w belongs to a finite number of $\mathcal{L}(X_m)$, so we conclude that $w \notin \bigcap_{M\in\mathbb{N}} \bigcup_{m\geq M} \mathcal{L}(X_m)$. We have shown that Item 2 is satisfied, considering its remaining inclusions are obvious.

Next, we modify the algorithm so that it produces a modified sequence $(X_{s(m)}, Y_{s(m)}, w_{s(m)})_{m\in\mathbb{N}}$, with $s\colon \mathbb{N} \to \mathbb{N}$ a nondecreasing computable function with $s(m) \leq s(m+1) \leq s(m)+1$ for all m. As its definition shows, in spite of the possibly misleading notation, $(s(m))_{m\in\mathbb{N}}$ is not an extracted subsequence of the mere $(m)_{m\in\mathbb{N}}$: it is that usual nondecreasing sequence of integers except some terms may be repeated. Any modified sequence of the form $(X_{s(m)}, Y_{s(m)}, w_{s(m)})_{m\in\mathbb{N}}$ with s obeying the above condition still satisfies the first two items of the lemma. Since the mixing distance and window size of X_m, the window size of Y_m, the length $|w_m|$ and the space used by the unmodified algorithm are all computable from m, we can choose s to grow slow enough so that the remaining conditions, Item 3 and Item 4, hold as well. □

Lemma 3.6. *Let $X \subset \mathcal{A}^{\mathbb{Z}}$ be a nonempty chain mixing Δ^0_2 subshift. Then there exists a sequence $(X_m, w_m)_{m\in\mathbb{N}}$, where each $X_m \subset \mathcal{A}^{\mathbb{Z}}$ is a mixing SFT and $w_m \in \mathcal{L}(X_m)$, such that the following conditions hold.*

1. *$\mathcal{L}(X) = \lim_{m\in\mathbb{N}} \mathcal{L}(X_m) = \bigcap_{M\in\mathbb{N}} \bigcup_{m\geq M} \{w_m\}$.*
2. *For each m, the window size and mixing distance of X_m and the length $|w_m|$ are all $o(\log m)$.*
3. *The function $m \mapsto (X_m, w_m)$ is computable in $O(2^m)$ space.*

Proof. Since X is Δ_2^0, there are two computable predicates φ_X^+ and φ_X^- such that $\mathcal{L}(X) = \{w \in \mathcal{A}^* \mid \forall k \; \exists \ell \; \varphi_X^+(w,k,\ell)\} = \{w \in \mathcal{A}^* \mid \exists k \; \forall \ell \; \neg\varphi_X^-(w,k,\ell)\}$. Consider the logical formula $\varphi_X(w,n)$ defined as follows.

1. Starting from $k = 0$, check for increasing $\ell \geq 0$ whether $\varphi_X^+(w,k,\ell)$ holds, and whenever it does, increment k and reset ℓ to 0. Do this until n pairs (k,ℓ) have been checked, and let k_+ be the final value of k.

2. Do the same for φ_X^- in place of φ_X^+, and let k_- be the final value of k.

3. Define $\varphi_X(w,n)$ as the truth value of $k_+ > k_-$.

Then for any $w \in \mathcal{L}(X)$, $\varphi_X(w,n)$ holds for all large enough n, while for $w \in \mathcal{A}^* \setminus \mathcal{L}(X)$, $\neg\varphi_X(w,n)$ holds for all large enough n.

We describe an algorithm that is very similar to that of Lemma 3.5. It stores a finite memory $M \subset \mathcal{A}^*$, which is initially empty. It proceeds in rounds, with round i consisting of the following steps.

1. Add a new word $w \in \mathcal{A}^*$ into M, in increasing order of length.

2. Let $Q = \{w \in M \mid \varphi_X(w,i)\}$, $F = M \setminus Q$ and $n = \max\{|w| \mid w \in M\}$. For each $w \in Q$, do the following. If there exists $|w| \leq p \leq n$ such that the SFT X_p defined by forbidding the words $F \cap \mathcal{A}^{\leq p}$ is mixing and satisfies $\mathcal{L}_j(X_p) = Q \cap \mathcal{A}^j$ for each $j \leq p$ and $w \in \mathcal{L}(X_p)$, choose the largest such p and output (X_p, w).

We claim that the sequence $(X_m, w_m)_{m \in \mathbb{N}}$ produced by the algorithm satisfies Item 1; the others follow as in Lemma 3.5. Given $k \geq 0$, let $i_0 \geq |\mathcal{A}|^k$ be so large that for all $v \in \mathcal{A}^{\leq k}$ and $i \geq i_0$, $\varphi_X(v,i)$ holds if and only if $v \in \mathcal{L}(X)$. Such an i_0 exists since $\varphi_X(v,i)$ converges to the correct value for each $v \in \mathcal{A}^{\leq k}$ separately and $\mathcal{A}^{\leq k}$ is a finite set. Then the SFT forbidding $F \cap \mathcal{A}^{\leq k}$ is precisely the SFT approximation $\mathcal{S}_k(X)$, which is mixing by assumption.

Suppose $i \geq i_0$ and consider an output (X_p, w) produced on step 2 of the algorithm on round i. We have $n = \max\{|w| \mid w \in M\} \geq k$. If $|w| \leq k$, then $w \in Q$ implies $w \in \mathcal{L}(X)$, and in this case $p \geq k$, since $|w| \leq p \leq n$, p is chosen as large as possible, and k is a valid choice. If $|w| > k$, then we have $p \geq k$ by definition. In either case, for each $j \leq k$ we have $\mathcal{L}_j(X_p) = Q \cap \mathcal{A}^j$ by definition of X_p; which is equal to $\mathcal{L}_j(\mathcal{S}_k(X)) = \mathcal{L}_j(X)$ since $i \geq i_0$. Thus we have shown $\mathcal{L}_j(X_p) = \mathcal{L}_j(X)$ for all $0 \leq j \leq k$ and all pairs (X_p, w) produced after round i_0. This implies $\mathcal{L}(X) = \lim_{m \in \mathbb{N}} \mathcal{L}(X_m)$.

Consider then $k \geq 0$ and a word $w \in \mathcal{L}_k(X)$. If $i \geq i_0$, then on step 2 of round i of the algorithm, $w \in Q$ and $p = |w|$ is a valid choice for w. Hence $w = w_m$ for infinitely many m's. On the other hand, for each $w \in \mathcal{A}^k \setminus \mathcal{L}(X)$ we have $w \notin Q$ for all $i \geq i_0$. Hence $w = w_m$ for only finitely many m's. This proves $\mathcal{L}(X) = \bigcap_{M \in \mathbb{N}} \bigcup_{m \geq M} \{w_m\}$. □

For the next lemmas, we recall some terminology from combinatorics on words. A set $C \subset \mathcal{A}^*$ is a *code*, if $c_1 \cdots c_m = c'_1 \cdots c'_n$ with $c_i, c'_i \in C$ implies $m = n$ and $c_i = c'_i$ for all $0 \leq i < m$. A word $w \in \mathcal{A}^*$ is *primitive* if $w = z^n$ with $z \in \mathcal{A}^*$ implies $n = 1$. A word or configuration $x \in \mathcal{A}^* \cup \mathcal{A}^{\mathbb{Z}}$ is *periodic with period* $p \geq 1$, if $x_i = x_{i+p}$ holds whenever both values are defined. The *conjugates* of $w \in \mathcal{A}^n$ are the words $w_{[i,n-1]} w_{[0,i-1]}$ for $0 \leq i < n$, and w is a *Lyndon word* if it is primitive and lexicographically minimal among its conjugates. Finally, w is *unbordered* if no prefix of w is a suffix of w.

Lemma 3.7. *Let $X \subseteq \mathcal{A}^{\mathbb{Z}}$ be an infinite mixing SFT with window size and mixing distance k, and let $W \subset \mathcal{L}(X)$ be finite. Denote $N = k(|W|-1) + \sum_{w \in W} |w|$. For any $n > 2N + 8k$, there exists a periodic configuration $x \in X$ with least period n such that $w \sqsubset x$ for all $w \in W$.*

Proof. We first prove that for each $m \geq 3k$, there exists an unbordered word $v \in \mathcal{L}(X)$ with $m \leq |v| < m + 3k$. Consider the width-k Rauzy graph G of X with edge labels in $\mathcal{A}$. Pick any vertex $p \in G$ and consider the set $C \subset \mathcal{L}(X)$ of first returns from p to itself, which is a code. Since k is a mixing distance for X, there exists $c \in C$ with $|c| \leq k$. Since X is infinite, there exists another first return $c' \in C$, which is either shorter than c, or satisfies $c'_i \neq c_i$ for some $0 \leq i < k$. In the first case we set $d = c'$, and in the latter we extend the prefix $c'_0 \cdots c'_i$ into a first return $d \in C$ with $|d| < 2k$. As C is a code, c and d are not powers of the same word. Then $c^\ell d^\ell \in \mathcal{L}(X)$ is primitive for all $\ell \geq 2$ [Lot97, Theorem 9.2.4], so one of its conjugates $v \in \mathcal{L}(X)$ is a Lyndon word, hence unbordered by [Lot97, Proposition 5.1.2]. The claim on $|v|$ holds for $\ell = \lceil m/|cd| \rceil$.

Denote $W = \{w_1, \ldots, w_{|W|}\}$. Since k is a mixing distance for X, there exist gluing words $u_1, \ldots, u_{|W|-1} \in \mathcal{L}_k(X)$ with $u = w_1 u_1 w_2 u_2 \cdots u_{|W|-1} w_{|W|} \in \mathcal{L}_N(X)$. Let $n > 2N + 8k$. Let $v \in \mathcal{L}(X)$ be an unbordered word with $n - N - 4k < |v| \leq n - N - 2k$. As k is also a window size for X, there exist gluing words $a \in \mathcal{L}_k(X), b \in \mathcal{L}_{n-|uav|}(X)$ with $x = {}^\infty(uavb)^\infty \in X$. Each $w \in W$ occurs in this configuration, since they occur in u. The least period of x is $|uavb| = n$, since v is unbordered and $|v| > n/2$. □

The construction of the unbordered word v in the above proof is essentially [BP09, Lemma 2]. We repeat it here, since we need finer control on the lengths of the words.

Lemma 3.8. *Let $n \geq 1$, and suppose that every word of length $2n$ occurring in $x \in \mathcal{A}^{\mathbb{Z}}$ is q-periodic for some $1 \leq q \leq n$ (which might depend on the word). Then x is q-periodic for some $1 \leq q \leq n$.*

Proof. Let $i \in \mathbb{Z}$ and $k \geq 2n - 1$. We prove by induction on k that $u = x_{[i,i+k]}$ is q-periodic for some $1 \leq q \leq n$. The claim follows when we let k grow and choose $i = -\lfloor k/2 \rfloor$.

The case $k = 2n - 1$ is true by assumption, so suppose $k \geq 2n$. Denote $u = vwa$, where $v \in \mathcal{A}^+$, $w \in \mathcal{A}^{2n-1}$ and $a \in \mathcal{A}$. Then vw is p-periodic and wa is q-periodic for some $p, q \leq n$. Recall the periodicity theorem of Fine and Wilf [FW65]: if a word has periods p and q, and length at least $p + q - GCD(p,q)$, then it also has $GCD(p,q)$ as a period. The word w satisfies the conditions because $p, q \leq n$ and $\gcd(p,q) \geq 1$, so w is $GCD(p,q)$-periodic.

The only element of proof left is that if vw is p-periodic, wa is q-periodic and w is $GCD(p,q)$-periodic, then vw and wa are $GCD(p,q)$-periodic. First, we have $a = w_{|w|-q}$; and since $w_{|w|-q} = w_{|w|-GCD(p,q)}$ by $GCD(p,q)$-periodicity, we have that wa is $GCD(p,q)$-periodic.

For vw, notice that for any $0 < i \leq p$ and $j \in \mathbb{N}_0$ so that the following makes sense, we have $v_{|v|-i-jp} = w_{p-i} = w_{p-i-GCD(p,q)} = v_{|v|-i-jp-GCD(p,q)}$; and so v is $GCD(p,q)$-periodic. Moreover, $v_{|v|-i} = w_{p-i} = w_{GCD(p,q)-i}$ by $GCD(p,q)$-periodicity of w; this completes the proof that vw is $GCD(p,q)$-periodic.

As a consequence, u itself is $GCD(p,q)$-periodic, which is what we claimed. □

Lemma 3.9. *Fix a finite alphabet $\mathcal{A}$. Given the Rauzy graph of a mixing SFT $X \subset \mathcal{A}^{\mathbb{Z}}$ with window size and mixing distance m, and two words $u, w \in \mathcal{L}_m(X)$, the time complexity of computing a gluing word $v \in \mathcal{A}^m$ such that $uvw \in \mathcal{L}(X)$ is at most $\exp(O(m))$.*

Proof. The nodes of the Rauzy graph G of X are words of length m, so its size is at most $|\mathcal{A}|^m$. Computing v amounts to finding a length-$2m$ path from u to v in G. We perform a breadth-first search, computing for each $i = 0, 1, \ldots, m$ the set of vertices $C_i \subset G$ that are reachable from u in exactly i steps, and the set D_i of vertices from which w is reachable in exactly i steps. Since the in- and outdegree of each vertex of G is at most $|\mathcal{A}|$, we have $|C_i|, |D_i| \leq |\mathcal{A}|^i$, and C_i and D_i can be computed in time $Poly(|\mathcal{A}|^m \cdot |\mathcal{A}|^i) = |\mathcal{A}|^{O(m)}$. We can choose any word in $C_m \cap D_m$ as v, and finding one takes another $|\mathcal{A}|^{O(m)}$ steps. □

3.3.3 Walls, counters and conveyor belts

The high-level structure of the CA f is the same for both cases of Theorem 4. We define the alphabet $\mathcal{B}$ of the CA f as a set larger than $\mathcal{A}$, which is the alphabet of the subshift we want to realize. The alphabet $\mathcal{B}$ consists of three layers as listed in the construction of Section 3.1 (using the letter $\mathcal{B}$ in place of $\mathcal{A}$): the Main Layer $\mathcal{B}_{\text{main}}$, the Computation Layer $\mathcal{B}_{\text{comp}}$, and the Cleaning Layer $\mathcal{B}_{\text{clean}}$. To define it, let $\mathcal{M}$ be a Turing Machine with state set Q, initial state $q_0 \in Q$, tape alphabet Γ, and blank tape symbol $\gamma_0 \in \Gamma$. We will describe the behavior of $\mathcal{M}$ later on; for now, we only need to name its components in order to define the alphabet of f.

The Main Layer has alphabet $\mathcal{B}_{\text{main}} = \{\mathcal{I}, \$, \mathcal{W}_\$\} \cup \{\mathcal{W}_a \mid a \in \mathcal{A}\} \cup \mathcal{A}$. By default, symbols of the subset $\mathcal{A}_1 := \mathcal{A} \cup \{\$\}$ are continually shifted to the left. The "decorated" wall symbols $\mathcal{W}_a$ for $a \in \mathcal{A}_1$ behave exactly like the $\mathcal{W}$-symbols of Section 3.1, and the decorations allow us to shift the symbols of $\mathcal{A}_1$ through the walls. This allows a segment to receive data from another segment on its right in order to determine whether they should merge. We identify with $\mathcal{A}$ the states $b \in \mathcal{B}$ such that $\pi_{\text{main}}(b) \in \mathcal{A}$ and whose other layers are blank. They will be the only states visible in the generic limit set, allowing for the realization of the desired subshift $X \subset \mathcal{A}^{\mathbb{Z}}$.

The Computational Layer of f consists of four sub-layers, called the Right Conveyor Belt Layer, Comparison Layer, Turing Machine Layer, and Clock Layer. It also contains a blank symbol, which we denote by $\#$. The layers are denoted $\mathcal{B}_{\text{comp}} = \mathcal{B}_{\text{belt}} \times \mathcal{B}_{\text{cmpr}} \times \mathcal{B}_{\text{TM}} \times \mathcal{B}_{\text{clock}} \cup \{\#\}$. The projection maps from the components of $\mathcal{B}_{\text{comp}}$ are undefined on $\#$. The sub-layers are defined as follows.

- The Right Conveyor Belt Layer $\mathcal{B}_{\text{belt}} = \mathcal{A}_1$ contains symbols from $\mathcal{A}$ and blank symbols. By default, it is continually shifted to the right. Together with the Main Layer, it forms "conveyor belts" on which circular words over $\mathcal{A}_1$ are cyclically shifted.
- The Comparison Layer $\mathcal{B}_{\text{cmpr}} = \mathcal{A}_1$ also contains symbols from $\mathcal{A}$ and blanks. By default, it is continually shifted to the left.
- The Turing Machine Layer $\mathcal{B}_{\text{TM}} = Q \cup \Gamma$ is used to simulate the machine $\mathcal{M}$.
- We use the Clock Layer $\mathcal{B}_{\text{clock}} = \{0, 1, 2, 3\}$ to implement a ternary counter that times certain actions of $\mathcal{M}$.

We define the CA f over the course of the next few sections. We begin by stating "default behaviors" of some of the layers, which may be overridden in special circumstances that we explicitly describe as such. Let $x \in \mathcal{B}^{\mathbb{Z}}$ be arbitrary, and denote $y = f(x)$.

1. If $\pi_{\text{main}}(x_1) = a \in \mathcal{A}_1$, or $\pi_{\text{main}}(x_1) \in \mathcal{W}_{\mathcal{A}_1}$ and $\pi_{\text{belt}}(x_0) = a \in \mathcal{A}_1$, or $\pi_{\text{comp}}(x_0) = \#$ and $\pi_{\text{main}}(x_1) = a \in \mathcal{A}_1$, then $\pi_{\text{main}}(y_0) = a$. This means the $\mathcal{A}$-part of the Main Layer is generally shifted to the left. If the right neighbor of a cell is a wall, the data is instead copied from the Conveyor Belt Layer of the cell itself, onto the Main Layer of the same cell. Finally, if the cell has blank Computation Layer but its right neighbor does not, then the data is copied from the Conveyor Belt layer of that neighbor.

2. If $\pi_{\mathrm{belt}}(x_{-1}) = a \in \mathcal{A}_1$, or $\pi_{\mathrm{comp}}(x_{-1}) = \#$ and $\pi_{\mathrm{main}}(x_0) = a \in \mathcal{A}_1$, then $\pi_{\mathrm{belt}}(y_0) = a$. This means the Conveyor Belt Layer is generally shifted to the right, and if the left neighbor of a cell has blank Computation Layer, the data is instead copied from the Main Layer of the cell itself, onto the Conveyor Belt Layer of the same cell.

3. Suppose $\pi_{\mathrm{cmpr}}(x_0)$ is defined. If $\pi_{\mathrm{cmpr}}(x_1) = a \in \mathcal{A}_1$, or $\pi_{\mathrm{main}}(x_1) = \mathcal{W}_a$, then $\pi_{\mathrm{cmpr}}(y_0) = a$. This means the Comparison Layer is generally shifted to the left, and if the right neighbor of a cell is a wall, its decoration is copied instead.

4. Suppose $\pi_{\mathrm{main}}(x_0) \in \{\mathcal{I}\} \cup \mathcal{W}_{\mathcal{A}_1}$. If $\pi_{\mathrm{main}}(x_1) = a \in \mathcal{A}_1$, then $\pi_{\mathrm{main}}(y_0) = \mathcal{W}_a$. Otherwise $\pi_{\mathrm{main}}(y_0) = \mathcal{W}_\$$. This means all walls copy their decorations from the Main Layer of their right neighbor.

5. If $\pi_{\mathrm{clock}}(x_{-1})$ is defined and not equal to 3, then $\pi_{\mathrm{clock}}(y_0) = c'$, and otherwise $\pi_{\mathrm{clock}}(y_0) = c'+1$, where $3' = 0$ and $c' = c$ for $c \in \{0, 1, 2\}$.

Items 1 and 2 imply that if x contains a length-n contiguous run of cells whose Computational Layer is not blank, and which is bordered by a wall on the right and any symbol $b \in \mathcal{B}$ with $\pi_{\mathrm{comp}}(b) = \#$ on the left, then the Main and Conveyor Belt Layers of these cells hold a circular word $w \in \mathcal{A}_1^{2n}$ that f continually rotates. We call such a run of cells a *conveyor belt*; each properly formatted segment will contain one. From the last part of Item 1, the symbols of w are also copied on the Main Layer of the left bordering cell, which will thus receive a periodic sequence of symbols $www\cdots$.

Items 3 and 4 imply that if a run of $\mathcal{A}_1$-cells on the Comparison Layer is bordered on the left by a wall, then that wall will capture the $\mathcal{A}_1$-symbols that are shifted toward it, and pass them to the Comparison Layers of the cells on its left. This flow of information is depicted in Figure 3.2.

The idea of item 5 is that the Clock Layers of a finite run of cells encode a ternary counter that a single application of f increments. The least significant digit is the leftmost one, and the state 3 denotes a 0 that holds a carry. Carries propagate to the right. The relevant property of the counter is the following.

Lemma 3.10. *Let $x \in \mathcal{B}^{\mathbb{Z}}$ be a configuration, and let $i \leq k \in \mathbb{Z}$ and $T \geq 0$ be such that for all $t \leq T$, $\pi_{\mathrm{clock}}(f^t(x_j)) \in \{0, 1, 2, 3\}$ for each $i \leq j \leq k$, but $\pi_{\mathrm{clock}}(f^t(x_{i-1}))$ is undefined. Then the sequence $(\pi_{\mathrm{clock}}(f^t(x))_k)_{t=0}^T$ is eventually periodic with transient part of length at most $k - i + 1$ and eventual period of length 3^{k-i+1}, and the state 3 occurs exactly once every 3^{k-i+1} steps in the periodic tail.*

Proof. The proof is done by induction on $k - i$. For $k = i$, the sequence has eventually periodic part $1, 2, 3, 1, 2, 3, \ldots$ which is reached after at most $1 = k - i + 1$ step (on which the state might be 0). For $k > i$, we know the sequence of digits at position $k - 1$ has eventual period of length 3^{k-i} and transient part of length at most $k - i$, and the state 3 occurs at position $k - 1$ exactly once in each period. The step after it does, the state at position k is incremented by one, and if its new value is 3, on the next step it resets to 0. On other time steps it retains its value. The claim follows. □

The idea of the construction is to write periodic configurations of (SFT approximations of) X onto the conveyor belts, which in turn feed them to $\mathcal{A}_1$-regions. All belts will eventually disappear from a generic configuration, leaving only the $\mathcal{A}_1$-regions whose contents approximate X in the generic limit set of f. The Comparison Layer captures this data through permeable walls, and the

Turing Machine Layer analyzes it in order to control the merge process of segments by comparing the contents of two adjacent segments.

The Cleaning Layer $\mathcal{B}_{\text{clean}}$ behaves exactly as in Section 3.1, dividing the initial configuration into non-overlapping segments. In particular, it retains the property that the outer signals s_o erase all non-s_o symbols they encounter, so that every segment initialized by $\mathcal{I}$-symbols is eventually fully formatted. When an $\mathcal{I}$-symbol becomes a (decorated) wall on the first time step, it also produces a simulated head of the machine $\mathcal{M}$ in state q_0 on the Turing Machine Layer of its left neighbor. In the following sections we describe how the machine performs computation and modifies the data on its segment.

3.3.4 Computation of periodic points

Under f, each formatted segment S goes through four different stages, in the following order: *computation stage*, *waiting stage*, *probe stage*, and *merge stage*. During the computation stage, the machine $\mathcal{M}$ computes and stores a periodic point of one of the SFTs given by Lemma 3.5 or Lemma 3.6. Once it is stored and continuously generated on the conveyor belt, the waiting stage begins. It lasts until the neighboring segment S' on the right of S has finished its computation stage. In the probe stage, the machine reads and analyzes the periodic point stored by S' to determine whether S should merge with it. Finally, in the merge stage the wall between the segments is erased and the periodic point of S is glued to the one of S'.

The four stages are mostly controlled by the simulated Turing Machine $\mathcal{M}$, which we now describe. It differs from a standard Turing Machine in several respects: we allow its head to move 0, 1 or 2 tape cells in one computation step, and to freely modify the contents of all cells in the vicinity of the simulated read-write head. Even though the simulated head is always on a cell that has a non-blank Computational Layer, it can modify the states of nearby cells that do not, in order to extend its computational tape (but it will never create new heads).

Recall that a simulated machine $\mathcal{M}$ is initialized on the right end of every properly initialized segment. We only describe the behavior of $\mathcal{M}$ in this context, as only the contents of properly initialized segments will be visible in the generic limit set of f – the rest is erased in finite time by s_o and s_i signals described in Section 3.1, here through the Cleaning Layer. Let thus S be a properly formatted segment in the f-trajectory of a configuration.

First, the head of $\mathcal{M}$ travels to the left end of the segment S, extending the Computational Layer. Then it measures the length ℓ of S, computes the largest power of two $2^m < \ell/2$, and erases the Computational Layer of the $\ell - 2^m$ leftmost cells of S. These cells will remain in $\mathcal{A}_1$-states from this point on, and the remaining 2^m cells of S will have non-blank Computational Layers until the segment merges with another one on its right. In particular, the machine $\mathcal{M}$ is now limited to $2^m = \Theta(\ell)$ tape cells. We call m the *rank* of the segment S. See Fig. 3.2 for a diagram of the structure of formatted segments.

Next, the machine $\mathcal{M}$ computes a word $u \in \mathcal{A}^{2^{m+1}}$ and stores it on the conveyor belt of the segment S. The definition of u is the first place where the two cases of the construction differ. In the $\Pi_1^0 \subset \Pi_2^0$ case, $\mathcal{M}$ computes the triple (X_m, Y_m, w_m) given by Lemma 3.5, which is doable in space 2^m if ℓ is large enough. Here X_m is a mixing SFT, $Y_m \subset X_m$ a nonempty SFT and $w_m \in \mathcal{L}(X_m)$ a word of length $o(\log m)$. The mixing distance and window size of X_m, and the window size of Y_m, are likewise $o(\log m)$. Denote by n_m the maximum of these numbers. In the Δ_2^0 case, $\mathcal{M}$ instead computes the pair (X_m, w_m) given by Lemma 3.6, and we denote by $n_m = o(\log m)$ the maximum of the mixing distance and window size of X_m. In both cases we may assume that the sequence

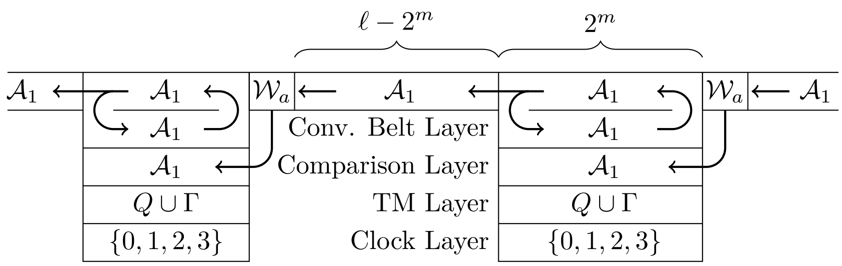


Figure 3.2: The anatomy of segments. Arrows indicate flow of information.

$(n_m)_{m\in\mathbb{N}}$ is nondecreasing and $n_m \to \infty$ as $m \to \infty$.

By Lemma 3.7, if ℓ is large enough, there exists a word $u \in \mathcal{A}^{2^{m+1}}$ such that the periodic configuration ${}^\infty u^\infty$ is in X_m, has least period 2^{m+1}, contains an occurrence of w_m, and in the $\Pi^0_1 \subset \Pi^0_2$ case, contains an occurrence of some word $v_m \in \mathcal{L}(Y_m)$ of length n_m. Indeed, we apply Lemma 3.7 to either $W = \{w_m\}$ or $W = \{w_m, v_m\}$, with v_m having negligible length compared to w_m.

In the case where ℓ is not large enough for all of the above, we use $u = a^{2^{m+1}}$ for an arbitrary $a \in \mathcal{A}$ instead.

The machine $\mathcal{M}$ computes such a u and writes it onto the conveyor belt. This concludes the computation stage of S.

Under the CA f, the word u is continually fed to the $\mathcal{A}_1$-cells on the left half of the segment S. From this point on, these cells will always hold $\mathcal{A}$-states, that are $\mathcal{A}_1 \setminus \$$-states.

3.3.5 Comparing periodic points

When the machine $\mathcal{M}$ has finished writing the word $u \in \mathcal{A}^{2^{m+1}}$ onto the conveyor belt of its segment S, it initiates the waiting stage by traveling to the right end of S. It waits there until the wall on its right stores an $\mathcal{A}$-state indicating that the segment S' directly to the right of S has finished its computation stage and stored some word $u' \in \mathcal{A}^*$ on its belt. Once this happens, the segment S enters the probe stage.

During the probe stage, the machine $\mathcal{M}$ will repeatedly capture a word occurring in the periodic point $x' = {}^\infty(u')^\infty$. Note that on each time step, the Comparison Layer of S now contains a length-2^m subword of x', which is continually shifted to the left and renewed through the wall between the two segments. The machine $\mathcal{M}$ waits on the rightmost cell of S until the Clock Layer of that cell contains a 3. We call this a *clock signal*, and by Lemma 3.10, it happens exactly once every 3^{2^m} time steps. Then the machine repeatedly stores four adjacent symbols from the Comparison Layer onto a single cell of its computation tape, waits for three steps, and takes one step to the left. Once it reaches the left end of the conveyor belt of S, its computation tape contains a word $v \in \mathcal{A}^{2^{m+2}}$ occurring in x' (that is, v is four times longer than the length of the computation tape). This process is illustrated in Fig. 3.3.

The machine $\mathcal{M}$ then checks whether the word v is q-periodic for some $q \leq 2^{m+1}$. If v is not q-periodic for any $q \leq 2^{m+1}$, we say $\mathcal{M}$ has *detected a merge candidate*. The idea is that we want to merge S with the segment S' only if S' has strictly higher rank, and detecting a merge candidate is evidence of this, since – with the exception of "false positives" mentioned later – having a larger

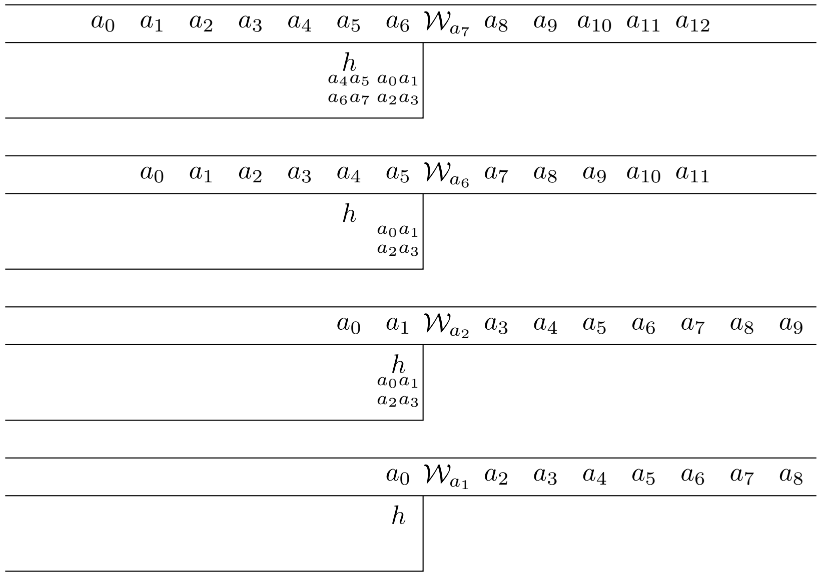


Figure 3.3: Capturing a word from the Comparison Layer. Time increases upward, possibly several steps at a time. Irrelevant layers and symbols are not shown. The letter h represents the head of the Turing Machine.

period for the word in S' means its conveyor belt itself was larger. Detecting one merge candidate is not enough: once $\mathcal{M}$ has performed this analysis, it erases v from its tape and starts over, waiting on the right end of the segment S for another clock signal. If ℓ is large enough, the capture, analysis and erasure of v takes less than 3^{2^m} computation steps, and by handling short segments separately (using specific local rules with big enough radius), we may assume this is the case for all ℓ. Thus $\mathcal{M}$ can start capturing a new word every time the clock signal occurs. The capturing process repeats until $\mathcal{M}$ has detected m merge candidates in total (not necessarily consecutively), after which S enters the merge stage. The reason for this is that if S' is produced by two short segments (of rank at most m) merging, right after this merge its Main Layer consists of two long words that can be repeated periodically, separated by a short period breaker word (see the merge process in Section 3.3.6). The machine $\mathcal{M}$ will detect at most $m-1$ such "false positive" merge candidates in the worst case, see Lemma 3.13.

Lemma 3.11. *Suppose the segment S' to the right of S has rank $m' > m$. Eventually either S enters the merge stage or S' merges with another segment on its right.*

Proof. Let $u' \in \mathcal{L}_{2^{m'+1}}(X_{m'})$ be the word stored on the conveyor belt of S'. By construction, the least period of the periodic point $x' = {}^\infty(u')^\infty \in X_{m'}$ is $2^{m'+1}$. The symbols of x are shifted to the left on the Main Layer of S', then through the wall separating S and S' onto the Comparison Layer of S. At each large enough time step t, if the segment S' has not yet merged with another segment on its right, the rightmost symbol of the Comparison Layer of S equals x'_{i+t} for some initial offset i. The clock signal of S arrives at time steps $t = j + n3^{2^m}$ for $n \in \mathbb{N}$ and some initial offset j, at which point the machine $\mathcal{M}$ simulated in S starts capturing a word of length 2^{m+2}, which thus equals $v(n) := x'_{[i+j+n3^{2^m}, i+j+n3^{2^m}+2^{m+2}-1]}$. Since $GCD(3^{2^m}, 2^{m'+1}) = 1$, we have

$\{v(n) \mid n \in \mathbb{N}\} = \{x'_{[n,n+2^{m+2}-1]} \mid n \in \mathbb{Z}\}$. If all of these subwords are periodic with period at most 2^{m+1}, then so is x' by Lemma 3.8, contradicting its construction. Hence at least one of the $v(n)$ is not periodic with a small period. When $\mathcal{M}$ captures this word, it detects a merge candidate, and when it has done so m times, S enters the merge stage. □

3.3.6 Merging segments

We now describe the merge stage of the segment S. Here the two cases differ more substantially. Recall that $n_m = o(\log m)$ is a window size for X_m and Y_m, and a mixing distance for X_m.

In the Δ_2^0 case, $\mathcal{M}$ captures a word $u \in \mathcal{A}^{n_m}$ from the Main Layer of S onto its computation tape, then rewrites the n_m symbols to the right of u on the Comparison Layer with \$-symbols, and finally captures another word $w \in \mathcal{A}^{n_m}$ from the Comparison Layer that occurs after the rewritten symbols. This process is controlled by some auxiliary markings $\mathcal{M}$ placed at the beginning of the merge stage; we omit the exact implementation details. See Fig. 3.4 for an illustration. We use the \$-symbols to mark the cells between u and w so that $\mathcal{M}$ can find them later; recall that the Main and Comparison Layers are continually shifted to the left by f.

In the $\Pi_1^0 \subset \Pi_2^0$ case, $\mathcal{M}$ waits for a clock signal before capturing the words u and w. Then it checks whether $w \in \mathcal{L}(Y_m)$, which takes $\exp(O(n_m)) = m^{o(m)}$ computation steps. If this is not the case, then $\mathcal{M}$ erases the words u and w from its tape, waits for another clock signal, and repeats the capturing process.

Next (immediately after capturing u and w in the Δ_2^0 case, and as soon as $w \in \mathcal{L}(Y_m)$ in the $\Pi_1^0 \subset \Pi_2^0$ case), $\mathcal{M}$ computes a *merge gluing word* $v \in \mathcal{A}^{n_m}$ as follows. In the $\Pi_1^0 \subset \Pi_2^0$ case, we simply require that $uvw \in \mathcal{L}(X_m)$. Such a word exists since $u \in \mathcal{L}(X_m)$ and $w \in \mathcal{L}(Y_m) \subset \mathcal{L}(X_m)$, and n_m is a mixing distance for X_m. By Lemma 3.9, $\mathcal{M}$ can compute v in $\exp(O(n_m)) = m^{o(m)}$ steps. In the Δ_2^0 case, $\mathcal{M}$ computes the largest integer $0 \leq d \leq n_m$ such that the length-d prefix $w_{[0,d-1]}$ occurs in the SFT approximation $X_{d,m} := \mathcal{S}_d(X_m)$. Note that each $X_{d,m}$ is also mixing with mixing distance n_m. Then it finds a v such that $uvw_{[0,d-1]} \in \mathcal{L}(X_{d,m})$, again in $m^{o(m)}$ steps.

The rest of the merge process is identical for the two cases. The machine modifies the conveyor belt of S by replacing the \$-symbols on the Comparison Layer with the symbols of v, one by one. As the \$-symbols are now within distance $m^{o(m)}$ from the right end of S and traveling left with constant speed, the ith symbol takes $\exp(O(i)) \cdot m^{o(m)}$ steps to replace, for a total of $\exp(m^{o(m)})$ steps. For large enough m we have $\exp(m^{o(m)}) \ll 2^m$, so there is enough time for $\mathcal{M}$ to perform these operations before the \$-symbols reach the left end of the conveyor belt of S (recall that $\mathcal{M}$ can move at speed 2 to catch up with the symbols moving at speed 1). Segments that are too short – that is, the resulting m is too small – are of bounded length. As such, they are handled separately through a dedicated automaton rule that merely erases them from the beginning. Consequently, we may assume that the previous merge process applies to all segments – that all of them are large enough. After this, the Main Layer of the conveyor belt contains a word of the form $auvb$, and the Comparison Layer contains cww', such that $|auv| = |c|$. We may arrange the copying process so that the simulated head of $\mathcal{M}$ ends up on top of the leftmost symbol of v.

Next, $\mathcal{M}$ travels left at speed 1 together with the Main and Comparison Layers. When it hits the left end of the conveyor belt, it turns back to the right and erases the Computation Layer of the segment S; it also rewrites the Main Layer with the contents of the Comparison Layer. Consequently, $auvww'$ ends up printed on the Main Layer, and the Computation Layer shrinks by "retracting" to the right of its segment. This process is illustrated in Fig. 3.5. When $\mathcal{M}$ reaches the right end of S, it erases itself and replaces the wall $\mathcal{W}_a$ with its decoration a as well. In this way, the segment S

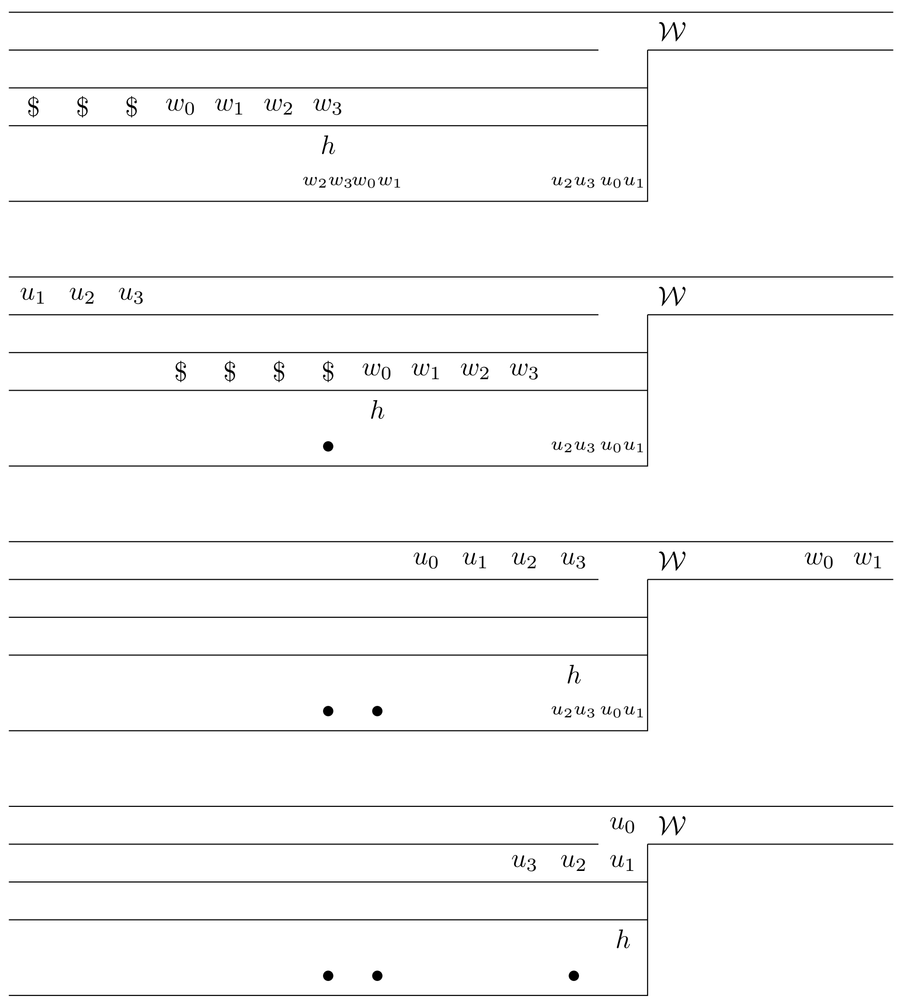


Figure 3.4: Capturing words at the beginning of the merge stage, illustrated with $n_m = 4$. Time increases upward several steps at a time. Irrelevant symbols and layers are not shown. The letter h represents the head of the Turing Machine. The dots are the auxiliary markings $\mathcal{M}$ has placed beforehand.

merges with its neighbor S' into one longer segment whose conveyor belt is identical to that of S'. This concludes the definition of f.

3.3.7 Proof of correctness

With f defined as above, we claim that its generic limit set is exactly X. Theorem 4 directly follows, since $f|_{\mathcal{A}^{\mathbb{Z}}} = \sigma|_{\mathcal{A}^{\mathbb{Z}}}$. Before that, we prove a few more lemmas about the behavior of segments under f.

Lemma 3.12. *Let S and S' be segments such that S' is the right neighbor of S with higher rank than S, and suppose S has entered the merge stage. Then eventually either S merges with S', or S' initiates its own merge process.*

Note that, *a priori*, S' might not have another segment on its right when it initiates the merge process.

Proof. In the Δ^0_2 case this is clear: once S enters the merge stage, it will capture u and w, compute the number d and the associated merge glue word v, and merge the segments.

Consider then the $\Pi^0_1 \subset \Pi^0_2$ case. Let $m < m'$ be the ranks of S and S', and let $u \in \mathcal{L}_{2^{m+1}}(X_m)$ and $u' \in \mathcal{L}_{2^{m'+1}}(X_{m'})$ be the words stored on their conveyor belts. By construction, u' has a length-$n_{m'}$ subword $w = w_{m'} \in \mathcal{L}(Y_{m'}) \subset \mathcal{L}(Y_m)$. As in the proof of Lemma 3.11, the simulated machine $\mathcal{M}$ in S repeatedly captures all of the subwords $\{x'_{[n,n+n_m-1]} \mid n \in \mathbb{Z}\}$ of length n_m in some order. Since $n_{m'} \geq n_m$, at least one of these words is a subword of w. Thus it occurs in Y_m and causes M to initiate the merge process, erasing the conveyor belt of S and the wall between S and S', and rewriting its Main Layer as described above, unless S' initiates its own merge process. □

Lemma 3.13. *Suppose a rank-m segment S has entered the merge stage. Then directly on its right there is another properly formatted segment of rank strictly above m.*

Proof. The proof is identical for the two cases from Lemma 3.12. It is enough to prove the result in the case where S has just entered the merge stage, since then we know that the rank of the segment directly to the right of S can only increase with time.

We proceed by induction on the time step $t \in \mathbb{N}$ on which S enters the merge stage. When S enters the merge stage, the simulated machine $\mathcal{M}$ in it has detected m merge candidates, which are captured words in $\mathcal{A}^{2^{m+2}}$ that are not q-periodic for any $q \leq 2^{m+1}$. Since S is a segment produced by the walls-and-counters construction, these words must originate from the conveyor belts of other segments. Thus, on some time step $t' < t$, there was another segment directly to the right of S. By the induction hypothesis, before time t that segment could only merge with properly formatted segments of higher rank. Since the number of time steps it takes to format a segment grows with its length, this implies that at time t there is also a segment S' – possibly not properly formatted – directly to the right of S. Let m' be its rank.

A merge candidate captured by S cannot be a subword of any periodic point stored in a conveyor belt of length at most 2^{m+1}. Thus, either it originates from a subword of a periodic point stored on a longer conveyor belt, or at least one of its symbols originates from a merge glue word. If the former condition holds for even one of the merge candidates, then the rank of the segment S'' directly to the right of S at some time step $t' < t$ was strictly above m. Since either $S' = S''$ or S' is produced by S'' merging with some segments to its right, which have even higher ranks by the induction hypothesis, we have $m' > m$.

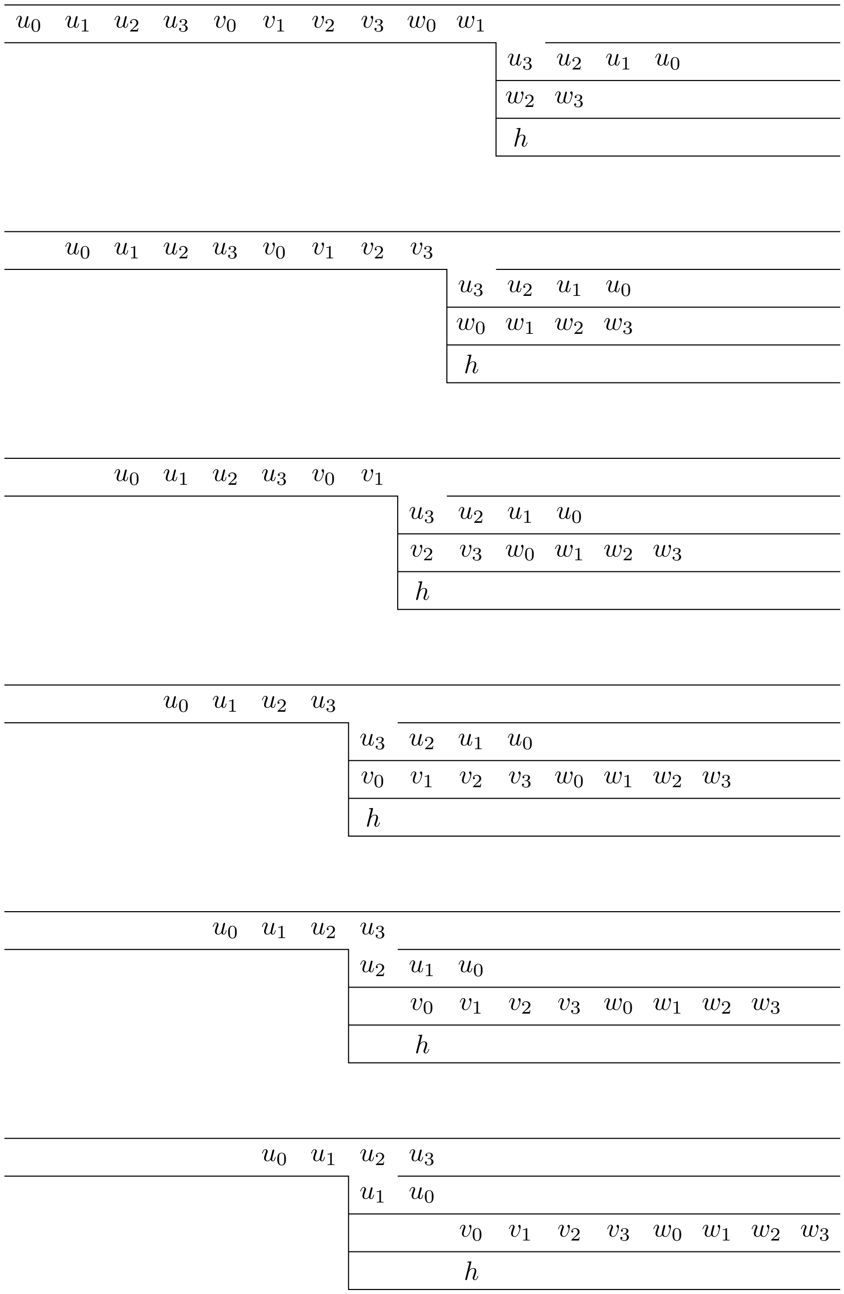


Figure 3.5: Erasing the Computation Layer, illustrated with $n_m = 4$. Time increases upward. Irrelevant states and layers are not shown. The letter h represents the head of the Turing Machine.

Suppose then that all m merge candidates contain symbols that originate from merge glue words. If any of these merge glue words is the result of a merge where one of the segments had rank above m, then $m' > m$ for the same reason as above. Suppose then that all of them result from merges between segments with rank at most m. Then the merge glue words have length at most $n_m = o(\log m)$. Since $\mathcal{M}$ starts the capture process only at clock signals, which occur every 3^{2^m} time steps, and $n_m < 2^{m+2} \ll 3^{2^m}$, two merge candidates cannot contain symbols originating from the same merge glue word – at least as long as m is big enough, and we can as always suppose that small rank cases are handled separately. Since a merge glue word appears uniquely with the merging of two segments, the segment S' is the result of at least $m+1$ segments eventually merging into one. Since the lowest possible rank is 1, by the induction hypothesis we have $m' > m$. □

Lemma 3.14. *For each $k \in \mathbb{N}$ there exists $m_k \in \mathbb{N}$ such that the following holds. Let $x \in \mathcal{B}^{\mathbb{Z}}$ be a configuration, $t \geq 0$, and $[i,j] \subset \mathbb{Z}$ an interval such that $S = f^t(x)_{[i,j]}$ is a segment of rank $m \geq m_k$ that has just entered the merge stage. Then for all $t' \geq t$ we have $f^{t'}(x)_{[j+1,j+k]} \in \mathcal{L}(X)$.*

Proof. Recall that $n_m = o(\log m)$ is a window size for X_m (and Y_m in the case where it exists), and a mixing distance for X_m. There exists $m_k \geq 0$ such that $\log_2 m_k > k$, $n_{m_k} \geq k$ and for all $\ell \geq m_k$, the SFT X_ℓ produced by Lemma 3.5 satisfies $\mathcal{L}_k(X_\ell) \subset \mathcal{L}_k(X)$, and the one produced by Lemma 3.6 satisfies $\mathcal{L}_k(X_\ell) = \mathcal{L}_k(X)$.

Fix $k \geq 0$. We prove by induction on t' that m_k has the required properties. Choose $m \geq m_k$, x, t and $[i,j]$ as in the claim. Lemma 3.13 implies that in $f^t(x)$, there is a segment S' of rank $m' > m$ directly to the right of S. Since $\log_2 m_k > k$, this segment's $\mathcal{A}$-part contains the interval $[j+1, j+k]$. Applying the same lemma repeatedly (whenever the segment containing cell $j+1$ merges with another one), we see that for all $t'' \geq t$, there is a segment of rank at least $m' > m_k$ in $f^{t''}(x)$ whose $\mathcal{A}$-part contains $[j+1, j+k]$.

Now, the word $r = f^{t'}(x)_{[j+1,j+k]}$ either originates from the conveyor belt of a segment of some rank ℓ, or at least one of its symbols originates from a merge gluing word $v \in \mathcal{A}^{n_\ell}$, computed as part of $uvw \in \mathcal{A}^{3n_\ell}$ during the merging of two segments of some ranks $\ell < \ell'$ (see Section 3.3.6). The same holds for each of the m merge candidates captured by S before time t. By Lemma 3.13, the last merge candidate originates from a segment of rank at least m, so we have $\ell \geq m \geq m_k$.

There are a few cases to consider.

- The word r originates from a conveyor belt. Then we have $r \in \mathcal{L}_k(X_\ell) \subset \mathcal{L}_k(X)$ by our choice of m_k.
- Some symbol of r originates from a merge gluing word and we are in the $\Pi^0_1 \subset \Pi^0_2$ case. Then $uvw \in \mathcal{L}(Y_\ell)$, hence $r \in \mathcal{L}_k(Y_\ell) \subset \mathcal{L}_k(X_\ell) \subset \mathcal{L}_k(X)$ by our choice of m_k.
- Some symbol of r originates from a merge gluing word and we are in the Δ^0_2 case. Then $uvw_{[0,d-1]} \in \mathcal{L}(\mathcal{S}_d(X_\ell))$ for the largest $0 \leq d \leq n_\ell$ with $w_{[0,d-1]} \in \mathcal{L}(X_\ell)$. The word $w_{[0,k-1]}$ was captured by the segment of rank ℓ before time step t'. By the induction hypothesis applied to that segment and our choice of m_k, we have $n_\ell \geq n_{m_k} \geq k$ and $w_{[0,k-1]} \in \mathcal{L}_k(X) = \mathcal{L}_k(X_\ell)$. It follows that $d \geq k$ since d is the largest possible integer so that $w_{[0,d-1]} \in \mathcal{L}(X_\ell)$ holds, and then $r \in \mathcal{L}_k(\mathcal{S}_d(X_\ell)) = \mathcal{L}_k(X_\ell) = \mathcal{L}_k(X)$.

In all cases we have $r \in \mathcal{L}(X)$. □

Proof of Theorem 4. We claim that the CA f constructed in this section satisfies $\widetilde{\omega}(f) = X$. The theorem follows from this, since $f|_{\mathcal{A}^{\mathbb{Z}}} = \sigma|_{\mathcal{A}^{\mathbb{Z}}}$ by construction, and $X \subset \mathcal{A}^{\mathbb{Z}}$.

Take any word $s \in \mathcal{L}(X)$. It occurs infinitely many times as w_m in the sequence of triples (X_m, Y_m, w_m) given by Lemma 3.5, or in the sequence of pairs (X_m, w_m) given by Lemma 3.6. Thus, in both cases of the theorem there are infinitely many different numbers ℓ such that a segment of length ℓ produced by the walls-and-counters part of f stores on its conveyor belt a periodic point that contains an occurrence of s.

We claim that the empty word enables s in the sense of Lemma 1.16. Take any cylinder set $[v]_i \subset \mathcal{B}^{\mathbb{Z}}$, where we may assume $|v| \geq 2|s|$ and $-|v| < i \leq 0$. Choose a large integer $k \in \mathbb{N}$ and consider the configuration $x = {}^\infty\$\mathcal{I}.v\mathcal{I}\$^k\mathcal{I}\$^\infty \in [v]_i$, where the dot denotes coordinate i. If k is large enough, the Cleaning Layer guarantees that $f^k(x)_{[i-1,i+|v|]}$ is a sequence of segments separated by walls, and Lemma 3.11 and Lemma 3.12 guarantee that the length-k segment on their right – the one that starts as $\mathcal{I}\$^k\mathcal{I}$ – will eventually merge with them all. This means that for large enough t, the word $f^t(x)_{[i-1,i+|v|+k+1]}$ is a single segment, and $f^t(x)_{[0,|s|-1]} \in \mathcal{A}^*$ lies in its $\mathcal{A}$-part. Due to the previous paragraph, we can find infinitely many k's – and consequently infinitely many x's in $[v]_i$ – so that the conveyor belt of the length-k segment contains an occurrence of s. This shows that the empty word enables s, hence $s \in \mathcal{L}(\widetilde{\omega}(f))$.

Conversely, let $s \in \mathcal{L}(\widetilde{\omega}(f))$ be arbitrary. By Lemma 1.16, some cylinder set $[v]_i \subset \mathcal{B}^{\mathbb{Z}}$ enables it. We may again assume that $|v| \geq 2|s|$ and $-|s| < i \leq 0$. Let $m = m_{|s|}$ be given by Lemma 3.14 for $k = |s|$, denote $\ell = 2^{m+1} + 1$ and consider the words $u = \$^n\mathcal{I}\$^\ell\mathcal{I}$ and $w = \mathcal{I}\$^{4(\ell+|v|)}\mathcal{I}\$^n$ for some large $n \in \mathbb{N}$. Since v enables s, we have that for infinitely many $t \in \mathbb{N}$, the cylinder set $C = [uvw]_{i-n-\ell-2}$ intersects $f^{-t}([s])$.

For all $x \in C$, the word $x_{[i-\ell-2,i+4(\ell+|v|)+1]}$ is a sequence of segments, the leftmost of which has rank m and the rightmost of which has the highest rank. Lemma 3.11 and Lemma 3.12 guarantee that as long as n is large enough (so that the left- and rightmost segments have no time to merge with any other segments), all these segments will eventually merge into one. Suppose this happens at time t. Let $t' < t$ be the time step at which a rank-m segment whose right wall is at coordinate i enters the merge stage. By our choice of $m = m_{|s|}$, we then have $f^{t''}(x)_{[i+1,i+k]} \in \mathcal{L}_k(X)$ for all $t'' \geq t'$ and $x \in C$. There exist $t''' \geq t + i + 1$ and $x \in C$ with $f^{t'''}(x)_{[0,k-1]} = s$ due to v enabling s. Then that word s has been shifted to the left during the $i + 1$ previous time steps, and combining this with $f^{t'''-i-1}(x)_{[i+1,i+k]} \in \mathcal{L}_k(X)$, we obtain that $f^{t'''-i-1}(x)_{[i+1,i+k]} = s \in \mathcal{L}_k(X)$. Therefore $s \in \mathcal{L}(X)$, which concludes the proof. $\square$

3.3.8 Bound optimality

Say that a CA $f\colon \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ is *eventually oblique* on an f-invariant subshift $X \subset \mathcal{A}^{\mathbb{Z}}$ if there exists $n \in \mathbb{N}$ such that for any $x \in X$, $f^n(x)_0$ depends only on coordinates of x in $(-\infty, -1]$ or $[1, \infty)$. The proof of [Tör20, Proposition 4] shows that if f is eventually oblique on $\widetilde{\omega}(f)$, then it satisfies Item 2 of Lemma 2.6. Hence we have the following.

Corollary 3.15. *If a CA $f\colon \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ is eventually oblique on $\widetilde{\omega}(f)$, then $\widetilde{\omega}(f)$ is inclusion-minimal.*

In particular, this holds if the restriction of f to $\widetilde{\omega}(f)$ is a (nonzero) power of the shift map. Notably, any generic limit set built with Theorem 4 is necessarily inclusion-minimal.

Remark 15. It is unclear whether Theorem 4 realizes all inclusion-minimal generic limit sets there are. Indeed, Proposition 2.7 gives a Π_2^0 upper bound on inclusion-minimal generic limit sets whereas we specifically realize the $\Pi_2^0 \supset \Pi_1^0$ and Δ_2^0 cases only.

In spite of the previous remark, we can deduce that the complexity bound of Proposition 2.7 is optimal. Indeed, there exist chain mixing subshifts with Π_2^0-complete languages: for example, take any Π_2^0-complete language $L \subset \{0,1\}^*$, and define the subshift $X \subset \{0,1,2\}^{\mathbb{Z}}$ by forbidding $2w2$ for each $w \in \{0,1\}^* \setminus L$. A given word $w \in \{0,1,2\}^*$ occurs in X if and only if it does not contain any of these forbidden words, as we can then extend it into the infinite configuration $\cdots 000w000 \cdots \in X$. Thus $\mathcal{L}(X)$ is Π_2^0-complete. This extendibility by 0-symbols also shows that X is chain mixing and contains a nonempty Π_1^0 subshift: the singleton $\{0^{\mathbb{Z}}\}$. As such, X can be built through Theorem 4, proving that the the complexity bound of Proposition 2.7 is optimal.

Corollary 3.16. *There exists a CA f with $f|_{\widetilde{\omega}(f)} = \sigma$ and $\widetilde{\omega}(f)$ an inclusion-minimal GLS, such that $\mathcal{L}(\widetilde{\omega}(f))$ is a Π_2^0-complete set.*

3.4 Corollaries

We can use Theorem 4 to characterize generic limit sets among several classes of subshifts. As it builds specific cellular automata that act as the shift map on their generic limit set, we show that the chain mixing assumption is unavoidable in that context: see Proposition 3.18 and Corollary 3.19. These results are straightforwardly deduced from the technical Lemma 3.17 [Aki93, p. 175] and Proposition 2.14 [Tör20, Prop. 6].

Lemma 3.17 (Page 175 of [Aki93]). *If (X,T) is a chain transitive topological dynamical system that is not chain mixing, then there is a factor map $\pi : (X,T) \to (F,S)$ onto a finite set F with at least two elements on which $S : F \to F$ is a cyclic permutation.*

Proposition 3.18. *Let $f\colon \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ be a CA such that $f|_{\widetilde{\omega}(f)} = \sigma|_{\widetilde{\omega}(f)}$. Then $\widetilde{\omega}(f)$ is a chain mixing subshift.*

Proof. By [Tör20, Prop. 5], $f|_{\widetilde{\omega}(f)} = \sigma|_{\widetilde{\omega}(f)}$ implies that $\widetilde{\omega}(f)$ is chain transitive. If it is not chain mixing, we obtain a contradiction from Lemma 3.17 and Proposition 2.14. □

This has the following corollary as consequence:

Corollary 3.19. *Let $X \subset \mathcal{A}^{\mathbb{Z}}$ be a one-dimensional chain transitive subshift that is either Π_2^0 and contains a nonempty Π_1^0 subshift, or is Δ_2^0. Then X is a generic limit set of some CA if and only if it is chain mixing.*

Proof. If X is chain mixing, then Theorem 4 implies that it can be realized as a generic limit set. Otherwise, Lemma 3.17 and Proposition 2.14 show that X is not a generic limit set. □

For one-dimensional SFTs (which all have computable languages), chain transitivity coincides with transitivity, and chain mixing with mixing. This gives a simple characterization of generic limit sets among transitive SFTs.

Corollary 3.20. *A one-dimensional transitive SFT is the generic limit set of a CA if and only if it is mixing.*

Additionally, we can completely characterize the generic limit sets among shift-minimal subshifts.

Corollary 3.21. *A one-dimensional shift-minimal subshift is the generic limit set of a CA if and only if it is chain mixing and Δ_2^0.*

Proof. Any one-dimensional shift-minimal chain mixing Δ_2^0 subshift can be realized as a generic limit set, due to Theorem 4.

Conversely, by Corollary 2.16, a shift-minimal generic limit set must be Δ_2^0; and it is a subshift by Proposition 1.23. All shift-minimal subshifts are in particular transitive due to Proposition 2.13, and so they are chain transitive. The result now follows from Corollary 3.19. □

Part II

Subshifts with Projective Restrictions

The first part of this thesis introduced the notion of subshifts in one dimension: they are sets of biinfinite words written with a finite alphabet, with the restriction of not containing any forbidden finite word from a (possibly infinite) given list. Endowed with the shift map that translates a configuration to the left, one-dimensional subshifts form an important, well-known class of discrete dynamical systems. Subshifts can also be defined in two dimensions: the configurations are described by symbols filling the discrete plane instead of a mere line, and they must not contain a list of forbidden two-dimensional patterns. Here, the shift action translates the configurations by two-dimensional integer vectors.

Over the years, several notions have been defined to try and characterize how complex it was to build subshifts in one or two dimensions. Periodicity [Ber66, Rob71, CI96, JR15, GS21] and pattern complexity [MH38, Cas00, CN10, KM21] are some instances. Here, we focus on two of the most well-known of these: entropy and the Domino Problem.

First, entropy, as its name inspired from physics and information theory suggests, is a quantity that (very broadly speaking) measures “how chaotic” a subshift can be. That is to say: when we try and build configurations from a given subshift, do we have many possibilities in the symbols we place on the line/plane? If we fix a finite pattern, does it determine an entire, infinite configuration, or do we have room for choices? In short, what leeway do we have to complete it into a valid element of the subshift? This is measured by looking at bigger and bigger window sizes (of the discrete line or the discrete plane), and listing how many valid patterns exist with such a size. The asymptotical behavior of this quantity has at most an exponential growth; we can measure how fast that growth is, and record the related exponent: the resulting positive real number is the entropy.

Second, the Domino Problem is about how easy it is to characterize the nonempty subshifts that can be built in one/two dimensions. Given an alphabet and a list of forbidden patterns, is it straightforward to determine whether at least one configuration will be valid with these? Of course, as humans cannot grasp high numbers of symbols or adjacency rules, what the Domino Problem really asks is whether it is possible to implement an algorithm that could tell apart the empty and the nonempty subshifts, provided only with their alphabet and the list of their forbidden patterns. There is, of course, always a way of getting an answer if the given subshift is empty: by testing any pattern of any size and shape, a computer will find a shape that cannot be filled whatever the choice of symbols. Such a semi-algorithm always ends up returning that an empty subshift is indeed empty, but will never tell a thing for nonempty subshifts, as it will run indefinitely. The question of the existence of a semi-algorithm listing the *nonempty* subshifts is consequently relevant.

As it turns out, both the entropy and the Domino Problem behave very differently depending on the setting: one dimension (over $\mathbb{Z}$) or two dimensions (over $\mathbb{Z}^2$). In dimension one entropies can only be the logarithm of Perron numbers, a specific, computable set of nonnegative real numbers. Meanwhile, in two-dimensional subshifts they require a higher computational power than usual Turing Machines: a famous result by Hochman and Meyerovitch [HM10] states that the possible values of the entropy for SFTs of dimension $d \geq 2$ are exactly the non-negative right-recursively enumerable (Π_1^0-computable, see Section 1.3.2) real numbers. On subshifts over $\mathbb{Z}$, the Domino Problem is known to be decidable, by equivalence with a problem of finding biinfinite paths on graphs. On those over $\mathbb{Z}^2$, Wang [Wan61] conjectured in the 60s that the Domino Problem was decidable too, and produced an algorithm of decision relying on the hypothetical fact that all subshifts of finite type (SFTs, subshifts with a finite number of forbidden patterns) contained some periodic configuration. However, his claim was disproved by Berger [Ber66] first, then through various methods by many others in the decades that followed [Rob71, Moz89, Kar96, Kar07, JR15], additionally proving that no other algorithm could work: the Domino Problem over $\mathbb{Z}^2$ is undecidable.

How can this difference in complexity be explained? An approach that is part of the motivation for Part III of the present thesis is to generalize subshifts to an even wider class of structures – discrete, countable groups – and to look for similar phenomena on them with the hope of getting a bigger picture on entropy [Bar19] or the Domino Problem [Jea15,CGS17,Coh17,ABJ18,ABM19]. In the same vein, generalizations have additionally been done on fractal (self-similar) structures to understand the Domino Problem better [BS16, Bar20]. Several papers from the past decade also explore how adding dynamical constraints to subshifts on $\mathbb{Z}^2$, such as the block gluing property [PS15, GdM19, GS21], minimality [GS18], or a restriction in the number of patterns of a given size [KM21] can have consequences on the entropy or the decidability of the Domino Problem, in some sense lowering the complexity of the available subshifts. This can be interpreted as a reduction of the computational power of the model as a whole under the added restriction.

The present part of this thesis focuses on a new approach to explore the frontier between decidability and undecidability that lies between the subshifts on $\mathbb{Z}$ and $\mathbb{Z}^2$. It is based on a simple observation: one-dimensional SFTs are easy to understand partly because they can be represented as directed graphs (see Section 1.2.3) while no such simple mathematical object can grasp forbidden patterns in dimension 2. Consequently, we can try and break down two-dimensional restrictions into one-dimensional ones. Of course, this is no miracle: most two-dimensional restrictions cannot be simply written off as a combination of horizontal and vertical one-dimensional forbidden patterns. However, this still yields a starting thought: if we start by fixing (well-understood, one-dimensional) horizontal restrictions in a two-dimensional subshift, and add two-dimensional forbidden patterns to that list, what are the available subshifts? This procedure, in some sense, echoes the longer-lasting idea of putting restrictions on projective subactions on $\mathbb{Z}$, that is: if we collapse two-dimensional subshifts into one-dimensional ones by looking only at the biinfinite horizontal words that appear, what is obtainable? The process of trying to realize specific subactions with an added dimension, called simulation, has a recent but rich history [Hoc16, DRS12, AS13].

Chapter 4 is dedicated to providing all the necessary formal notions evoked in this introduction. Then, in Chapter 5 and Chapter 6 we turn the point of view from simulation theorems around: instead of aiming to build a specific two-dimensional subshift to realize a precise projective subaction, we fix horizontal constraints beforehand and look at the diversity of available two-dimensional SFTs respecting these constraints – and possibly additional, two-dimensional ones. In few words, we study the accessible subsystems of a given two-dimensional SFT X (defined only by horizontal constraints): the only properties we known about them are that their entropy is bound by $h(X)$ (indirectly through [JKM07]) and a more general result by A. Desai [Des06] – recently extended to SFTs over countable amenable groups [BMP22] – proving that the set of accessible entropies is actually dense in the interval $[0, h(X)]$.

Here, we focus on the following sets, where the notation $X_{\mathcal{F}}$ refers to a two-dimensional SFT with $\mathcal{F}$ as its list of forbidden patterns. Given a subshift of finite type H on one-dimension – with a finite list of forbidden patterns, that can be considered in two dimensions and written $\mathcal{H}$ as such – we study:

- the set $DP_h(\mathcal{H}) = \{< \mathcal{F} > \mid \mathcal{F} \text{ is finite and } X_{\mathcal{H}\cup\mathcal{F}} \neq \varnothing\}$ where $< \mathcal{F} >$ is the encoding of the finite set $\mathcal{F}$ of forbidden patterns;

- the set $\{h(X_{\mathcal{H}\cup\mathcal{F}}) \mid \mathcal{F} \text{ finite set of forbidden patterns}\}$, that we know is dense in the interval $[0, h(H)]$ by [Des06].

In Chapter 5, as published in a joint article with M. Sablik [ES22], we reach a full characterization:

- the set $DP_h(\mathcal{H})$ is decidable if and only if H contains only eventually periodic configurations (Theorem 9);
- for any SFT H, $\{h(X_{\mathcal{H}\cup\mathcal{F}}) \mid \mathcal{F}$ set of forbidden patterns$\}$ is exactly the set of Π_1^0-computable numbers in $[0, h(H)]$ (Theorem 11).

These results are strong: the first one means that apart from trivial horizontal restrictions, all of them lead to an undecidable Domino Problem under horizontal constraints H. The second one proves that any real number that could be the entropy for a subsystem with H as horizontal constraints – as they have to be Π_1^0-computable numbers and with an entropy smaller than H's – is indeed realized by some subsystem. A consequence of it, obtained by taking H as the full shift $\mathcal{A}^{\mathbb{Z}}$ (no forbidden pattern) for some alphabet $\mathcal{A}$, is that the set of possible entropies of two-dimensional SFTs on $\mathcal{A}$ is exactly the Π_1^0-computable numbers of $[0, \log(\mathcal{A})]$. The result by M. Hochman and T. Meyerovitch [HM10] does not answer this precise question because their construction can use an arbitrarily large number of letters.

In Chapter 6, we consider again two-dimensional subshifts parametrized by pre-fixed horizontal constraints, but we force the additional local rules to be vertical – contrary to Chapter 5 where they could have arbitrary shape. This specific question is interesting notably because a classical result among simulation theorems is that every effective one-dimensional subshift can be realized as the projective subaction on $\mathbb{Z}$ of a two-dimensional subshift of finite type [AS13]; however, in these constructions, the dynamic on the other direction – that is, the one-dimensional configurations we witness vertically – is trivial. Therefore, here we ask which one-dimensional, vertical subshifts can be compatible with a given horizontal dynamic, in the form of what [LMP13] calls an axial product. Moreover, considering such notion of compatibility between horizontal and vertical constraints is in line with the main thought behind the present part: breaking down two-dimensional constraints into easier to study, one-dimensional ones. We stated earlier that this process of separating horizontal and vertical restrictions would not work on all two-dimensional constraints; but it is interested to study the ones where it does work.

As such, in Chapter 6, given two one dimensional subshifts of finite type H and V we ask if there exist a two dimensional configuration where all horizontal lines are in H and vertical lines in V. Its results are from a joint conference article with N. Aubrun and M. Sablik [AES20], extended in [ES22].

The possibility of adding only vertical patterns is more complicated than the wider range of arbitrary additional forbidden patterns, because we need to understand how to transfer information – to build complicated configurations – with few interplay between the two directions. As a consequence, in Chapter 6, we restrict ourselves to horizontal constraints between nearest-neighbor symbols. We prove that if the corresponding SFT H satisfies a certain set of conditions, it is possible to simulate any two-dimensional SFT in some sense (Theorem 12).

With this result, we have a characterization of the nearest-neighbor one-dimensional SFTs which have undecidable Domino Problem with interplay DP_I (Theorem 13). For the entropy, we obtain a partial characterization; yet surprisingly we find horizontal subshifts H that can only have a decidable Domino Problem when vertical constraints are added, but that can realize two-dimensional SFTs with "strictly" Π_1^0-computable numbers as entropies (that is, numbers that are Π_1^0-computable but not merely computable).

Chapter 4

Two-Dimensional Subshifts and the Domino Problem

4.1 Symbolic Dynamics

This section heavily uses notions from Section 1.2.1 and assumes they are known. It is recommended to read the latter before the present one.

In what follows we define objects similar to their one-dimensional counterparts (on $\mathbb{Z}$), except they are now two-dimensional (using $\mathbb{Z}^2$).

Remark 16. All the notions that follow can also be defined for any $\mathbb{Z}^d, d > 0$ mutatis mutandis (mainly replacing any 2 with a d instead). However, they will only be of use on $\mathbb{Z}$ and $\mathbb{Z}^2$ in the present part, hence this broader d-dimensional version will not be detailed here.

See Chapter 7 for an even more general setting on any finitely generated group.

As before, we consider an alphabet $\mathcal{A}$.

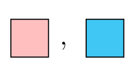
Definition 4.1. Let $\mathcal{A}$ be an alphabet. Endow $\mathcal{A}^{\mathbb{Z}^2}$ with the prodiscrete topology t_π.

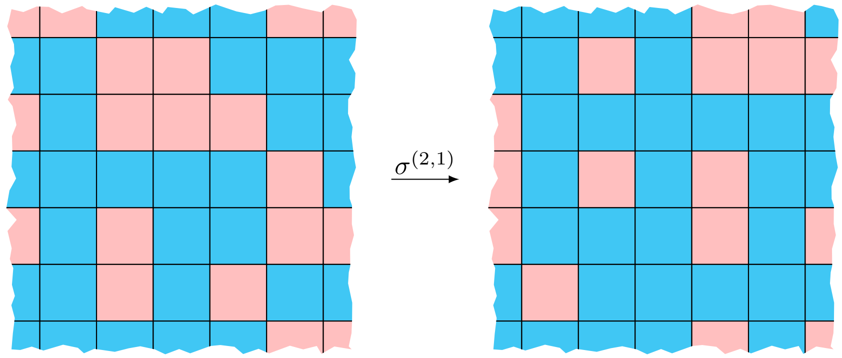
For any $\vec{v} \in \mathbb{Z}^2$ define the *shift map* $\sigma^{\vec{v}} : \mathcal{A}^{\mathbb{Z}^2} \to \mathcal{A}^{\mathbb{Z}^2}$ such that $\sigma^{\vec{v}}(x)_{\vec{k}} = x_{-\vec{v}+\vec{k}}$. It defines a natural $\mathbb{Z}^2$ action σ on $\mathcal{A}^{\mathbb{Z}^2}$.

$(\mathcal{A}^{\mathbb{Z}^2}, t_\pi, \sigma)$ is called the *two-dimensional full shift* over $\mathcal{A}$, and is a compact space.

Any $x \in \mathcal{A}^{\mathbb{Z}^2}$, called a *configuration*, can be seen as a function from $\mathbb{Z}^2$ to $\mathcal{A}$ and we write $x_{\vec{k}} := x(\vec{k})$.

Example 15. In the picture that follows, we illustrate an alphabet $\mathcal{A}$, a local window of a specific configuration from $\mathcal{A}^{\mathbb{Z}^2}$, and the effect of the shift map σ applied with vector $(2, 1)$.

$\mathcal{A} = \{$  $\}$



Definition 4.2. A *two-dimensional subshift* X is a σ-invariant closed subset of $\mathcal{A}^{\mathbb{Z}^2}$.

Just as in Definition 1.17, in the general framework of dynamical systems, a subshift is just a subsystem of a full shift.

Since two dimensions do not allow the use of words as in Section 1.2.1, we define a broader notion:

Definition 4.3. A *pattern* p is a finite configuration $p \in \mathcal{A}^{P_p}$ where $P_p \subset \mathbb{Z}^2$ is finite.

We say that a pattern $p \in \mathcal{A}^{P_p}$ *appears* in a configuration $x \in \mathcal{A}^{\mathbb{Z}^2}$ – or that x *contains* p – if there exists $\vec{v} \in \mathbb{Z}^d$ such that for every $\vec{\ell} \in P_p$, $\sigma^{\vec{v}}(x)_{\vec{\ell}} = p_{\vec{\ell}}$. We denote it $p \sqsubset x$.

This implies a property analogous to Property 1.18:

Property 4.4. *Any two-dimensional subshift X can be described by a set $\mathcal{F}$ of patterns called the set of* forbidden patterns*. This means that the following holds:*

$$X = \{x \in \mathcal{A}^{\mathbb{Z}^2} \mid \forall p \in \mathcal{F}, p \not\sqsubset x\}.$$

Just as in the one-dimensional case, it is often denoted $X_{\mathcal{F}}$ instead of merely X.

Remark 17. Once again, several sets of forbidden patterns can define the same subshift.

The conjugacy invariant notions of being a SFT, a sofic or an effective subshift are exactly the same, with the same implications, as in Property 1.22.

Definition 4.5. For a two-dimensional subshift X, the set of all patterns of size $n_1 \times n_2$ (format "width $\times$ height") that appear in configurations of X is denoted by $\mathcal{L}_X(n_1, n_2)$, and its cardinality by $N_X(n_1, n_2)$.

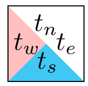
For a one-dimensional subshift H, we write $\mathcal{L}_H = \cup_n \mathcal{L}_H(n)$.

The following definition is the 2D version of Definition 1.25.

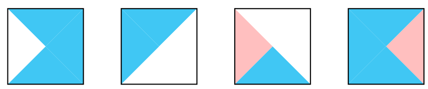
Definition 4.6. A two-dimensional SFT is said to be *nearest-neighbor* if it has a list of forbidden patterns so that for any such forbidden pattern p, $P_p = \{(0,0),(0,1)\}$ or $P_p = \{(0,0),(1,0)\}$.

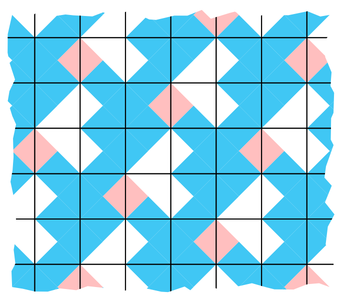
The most well-known nearest-neighbor two-dimensional SFTs are the *Wang shifts*, defined by a finite number of square tiles (called Wang tiles) with colored edges that must be placed with matching colors. Formally, these tiles are quadruplets of symbols (t_e, t_w, t_n, t_s). A Wang shift is described by a finite *Wang tile set*, an alphabet made of such quadruplets; and local rules $x(i,j)_e = x(i+1,j)_w$ and $x(i,j)_n = x(i,j+1)_s$ for all integers $i, j \in \mathbb{Z}$.

A Wang tile looks like the following picture, where the quadruplet (t_e, t_w, t_n, t_s) is most of the time replaced by colors. We will often re-use colors between the North-South axis and the East-West axis symbols with no consequence, as Wang tiles cannot be rotated.



Example 16. An example of Wang tile set and a tiling – a configuration of the associated Wang shift – is given below.

$\mathcal{A} = \{$  $\}$



Remark 18. Not all nearest-neighbor shifts are Wang shifts in a transparent way where you could keep the same alphabet. Consider for instance the subshift with alphabet $\mathcal{A} = \{0, 1\}$ and as only rules no vertical constraints, and the following horizontal restrictions: the only symbols allowed to follow each other are 00, 01 and 10. The symbols 0 and 1 cannot be replaced Wang tiles, because if they were, using the quadruplet notation from above, we would have:

- 00 being authorized means $0_w = 0_e$;
- 01 being authorized means $0_e = 1_w$;
- 10 being authorized means $1_e = 0_w$.

Then $1_e = 1_w$ and so 11 should be authorized, and this is not the case.

In spite of the previous remark, we have the following property, as a step further from Property 1.30, that also holds in two dimensions:

Property 4.7. *Any two-dimensional SFT is conjugated to a Wang shift.*

Remark 19. Notably, a two-dimensional SFT is empty if and only if the corresponding Wang shift is – that fact will allow us to focus on Wang shifts only if need be.

4.2 Two major tools for studying subshifts

4.2.1 Domino Problem

The present subsection defines a fundamental question – a decision problem, see Definition 1.38 in symbolic dynamics: is there an algorithm that, taking as input any finite list of forbidden patterns, can tell if the associated SFT is empty?

Definition 4.8. For $d = 1$ and $d = 2$, define $DP(\mathbb{Z}^d)$, the *Domino Problem in dimension* d, as the decision problem on the set of all finite lists of forbidden patterns in $\mathcal{A}^{\mathbb{Z}^d}$, asking for such a list $\mathcal{F}$, and answering Yes if the corresponding $X_{\mathcal{F}}$ is nonempty, and No otherwise.

Remark 20. Said otherwise, $DP(\mathbb{Z}^d)$ asks if $\{\langle\mathcal{F}\rangle \mid X_{\mathcal{F}}$ is a nonempty SFT$\}$ is decidable, where $\langle\mathcal{F}\rangle$ is an encoding of the set of forbidden patterns $\mathcal{F}$ suitable for a Turing Machine.

This decision problem has been intensively studied since Wang's pioneering works [Wan61], and two major results are well-known in symbolic dynamics:

Theorem 5. *$DP(\mathbb{Z})$ is decidable.*

Proof. The proof is based on Section 1.2.3: one can reduce the study of any SFT to its Rauzy graph. Notice how two conjugated subshifts are both SFTs if and only if one is; how both are nonempty if and only if one is; and how a configuration in a nearest-neighbor SFT is equivalent to a biinfinite path in its Rauzy graph. Although these are trivial statements on their own, they mean trying to answer the question of the present decision problem is purely equivalent to a study of graphs and biinfinite paths.

With what was written, a configuration in the original SFT is thus equivalent to finding a biinfinite path of vertices in the Rauzy graph. In finite oriented graphs, that is equivalent to finding a cycle. With this reasoning, we reduce $DP(\mathbb{Z})$ to the problem of finding a cycle in a finite graph, which is easily algorithmically decidable. □

Theorem 6. *(see for instance [Ber66, Rob71, Moz89, Kar07]) $DP(\mathbb{Z}^2)$ is undecidable.*

Remark 21. In some sense, the above result encapsulates how two dimensions allow for more "complicated" SFTs: basically, any one-dimensional SFT can be somewhat easily understood from its list of fordbidden patterns; while two-dimensional SFTs have enough room to recreate the computations of any Turing Machine – the reduction of $DP(\mathbb{Z}^2)$ to the Halting Problem being a key element in most proofs [Ber66, Rob71, Moz89] of its undecidability.

4.2.2 Entropy

One of the most important notions in symbolic dynamics is the entropy, which measures "how many" configurations exist in a subshift. Of course, this is somewhat philosophical: in most cases, a subshift contains an infinite number of configurations. The question is then how this infinity can be quantified. The main idea behind that is to study larger and larger pattern sizes (with growing n-length words or growing $n \times n$ squares) and count how many valid patterns exist for a given size. The asymptotic behavior of that quantity when n goes to infinity represents, in some sense, the number of configurations of the subshift.

Now, the full shift on alphabet $\mathcal{A}$ has $|A|^{n^d}$ valid patterns for a window of appropriate size ($n \times n \times \cdots \times n$, with d the dimension and the number of n's). That is,

$$N_{\mathcal{A}^{\mathbb{Z}^d}}(n, \ldots, n) = 2^{n^d \log_2(|A|)}.$$

That factor of the exponential growth that multiplies n^d is what we are interested in (up to possibly choosing a different logarithm), because it is what quantifies the growth rate. Of course, subshifts X in general with alphabet $\mathcal{A}$ will have a wider range of behaviors than the full shift, but since the asymptotic behavior of their number of patterns $N_X(n, \ldots, n)$ is bounded from above by the full shift's, we can also study that growth rate factor.

This reasoning is enough to naturally introduce the following crucial quantity:

Definition 4.9. Let X be a d-dimensional subshift. The *(topological) entropy* of X is

$$h(X) = \lim_{n \to \infty} \frac{\log_2(N_X(n, \ldots, n))}{n^d} = \inf_n \frac{\log_2(N_X(n, \ldots, n))}{n^d}.$$

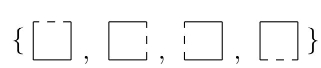
The entropy is an inf that is also a limit due to Fekete's Lemma (see for instance [LM95, Lemma 4.1.7] for a version of it).

Proposition 4.10. *(see [LM95, Cor. 4.1.9, Cor. 4.1.10]) If X and Y are conjugate, then $h(X) = h(Y)$.*

If Y is a factor of X, then $h(X) \geq h(Y)$.

Remark 22. Some subshifts will obviously have a sub-exponential growth in the number of patterns for n-sized windows (consider for instance a subshift with only a periodic configuration and its translates), which will lead to their entropy being 0. Though finer invariants than the entropy can be defined with more precise estimations of the growth rate, it is already an important quantity to quantify which subshifts allow for numerous configurations (high entropy) and which are very "rigid" in the patterns they allow (null entropy).

Example 17.

- Following the reasoning above, $h(\mathcal{A}^{\mathbb{Z}^d}) = \log_2(|A|)$.
- The one-dimensional subshift on alphabet $\mathcal{A} = \{0, 1\}$ defined as X_{11} is of entropy $h(X_{11}) = \log_2(\frac{1+\sqrt{(5)}}{2})$ [LM95, Ex. 4.1.6].
- The Robinson subshift [Rob71] illustrated in the introduction of Part III and often used as the baseline of proofs for the undecidability of the Domino Problem over $\mathbb{Z}^2$, has zero entropy. An easy argument behind this is how any $n \times n$ square of tiles is entirely determined by the tiles on its $4(n-1)$ perimeter.
- The Kari-Culik subshift [CI96,Kar07], another subshift used to prove the undecidability of the Domino Problem over $\mathbb{Z}^2$, is of nonzero entropy [DGG14].
- The dimer subshift given by the alphabet 

  where dashed borders must meet dashed borders, has entropy

$$h = \frac{1}{4} \int_0^1 \int_0^1 \log\left(4 + 2\cos(2\pi x) + 2\cos(2\pi y)\right) dx dy$$

due to the computation being similar to the one of perfect matchings of dimers, on which one can use a theorem by Kasteleyn notably detailed in [Ken02]. This example serves as a proof that subshifts even simple in appearance and description can have a highly nontrivial entropy.

There is actually quite a difference between the real numbers accessible as entropies for one-dimensional SFTs and for two-dimensional SFTs. The following theorem dated from 1984 [Lin89] and notably stated in the comprehensive [LM95] describes the entire range of entropies in one dimension.

Theorem 7 (Th. 4.3.3 of [LM95])**.** *The set of all entropies accessible to one-dimensional SFTs is exactly the following set of real numbers:*

$$\left\{\frac{p}{q}\log(\lambda)|p \in \mathbb{N}_0, q \in \mathbb{N}, \lambda \textit{ is a Perron number}\right\}$$

where a Perron number is any algebraic real number λ so that $\lambda > 1$ and for any other root α of their minimal polynomial, $\lambda > |\alpha|$.

Definition 4.11. A real number is *computable* if there exists a Turing Machine that can compute an approximation of it with any precision, when given said precision as input.

A real number is Π_1^0-*computable* – sometimes called *right-recursively enumerable* or r.r.e. – if there exists a decreasing computable sequence of rationals which converges toward that number.

Said convergence is from “the right” in the line of real numbers, hence the name of right-recursively enumerable. The other name is due to the ties this notion has with Π_1^0 countable sets from Section 1.3.2.

Remark 23. Any computable real number is Π_1^0, but there are Π_1^0 numbers that are not computable. A classic instance of this can be obtained with an enumeration $(\mathcal{M}_n)_{n\in\mathbb{N}}$ of all Turing Machines (see Definition 1.36): consider the real number $r = 0.\dots$ where the digit in position n is 0 if the Turing Machine $\mathcal{M}_n$ halts on the empty input, and 1 otherwise. As more and more precise lists of halting Turing Machines can be obtained algorithmically, we can obtain upper approximations of that real number, that converge toward it. But any ability to give a lower converging sequence, and thus an approximation with any precision of r, is fruitless due to Theorem 2. Consequently, r is Π_1^0 but not computable.

Another classic instance is the one of Chaitin omega numbers: for a given universal Turing Machine, its Chaitin number represents the probability that said Turing Machine will halt on a randomly generated input.

The characterization of the entropies in dimension 2 is due to Hochman and Meyerovitch [HM10]:

Theorem 8 (Th. 1.1 of [HM10])**.** *The set of all entropies accessible to two-dimensional SFTs is exactly the nonnegative Π_1^0 real numbers.*

4.3 Horizontal constraints

Considering how wide the gap is between the behaviors on $\mathbb{Z}$ and $\mathbb{Z}^2$, both in terms of Domino Problem and entropy, we want to understand how this difference arises. This study, done in Chapter 5, is based on fixing a “horizontal” one-dimensional subshift and completing it to make it two-dimensional some way or another. This gives rise to the following subsections dedicated to new concepts.

4.3.1 New definitions

Adding a dimension to a one-dimensional SFT H, the local rules allow to define a two-dimensional SFT where each line is a configuration of H chosen independently – hence the use of the letter H for the subshift that ultimately encodes the *horizontal* rules. We want to study the consequence of adding extra rules to that subshift: the most natural way of doing so is to add forbidden patterns – which can be two-dimensional. This is formalized in the next definition.

Definition 4.12. Let $H \subset \mathcal{A}^{\mathbb{Z}}$ be a one-dimensional SFT and $\mathcal{F}$ be a finite set of two-dimensional forbidden patterns. The two-dimensional SFT

$$X_{H,\mathcal{F}} := \{x \in \mathcal{A}^{\mathbb{Z}^2} \mid \forall j \in \mathbb{Z}, (x_{k,j})_{k\in\mathbb{Z}} \in H \text{ and } x \in X_{\mathcal{F}}\}$$

is called the subshift $X_{\mathcal{F}}$ *with added horizontal constraints from H.*

Remark 24. The projection of the horizontal configurations that appear in $X_{H,\mathcal{F}}$ (or, similarly, $X_{H,V}$ defined below) does not necessarily recover all of H. Indeed, some of the configurations in H will not necessarily appear in $X_{H,\mathcal{F}}$, because they may not be legally extended into a valid two-dimensional configuration.

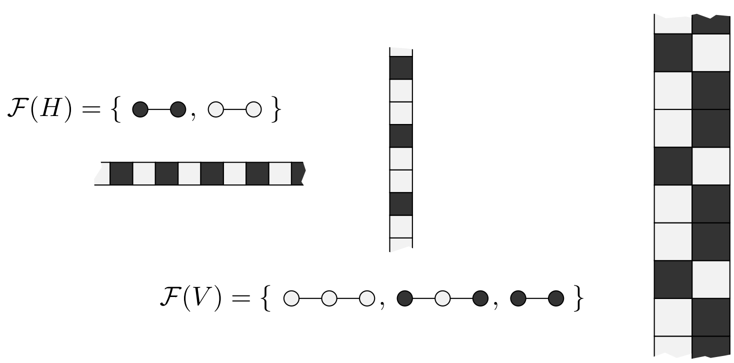
Another point of view is to search if two one-dimensional subshifts can be combined into a two-dimensional subshift where the first one appears horizontally and the second one vertically.

Definition 4.13. Let $H, V \subset \mathcal{A}^{\mathbb{Z}}$ be two one-dimensional SFTs. The two-dimensional subshift

$$X_{H,V} := \{x \in \mathcal{A}^{\mathbb{Z}^2} \mid \forall i, j \in \mathbb{Z}, (x_{k,j})_{k\in\mathbb{Z}} \in H \text{ and } (x_{i,\ell})_{\ell\in\mathbb{Z}} \in V\}$$

is called the *combined subshift* of H and V, and uses H as horizontal rules and V as vertical rules.

Example 18. Choose $\mathcal{A} = \{0, 1\}$, H nearest-neighbor and forbidding 00 and 11, and V forcing to alternate a 1 and two 0s: the resulting $X_{H,V}$ is empty, although neither H nor V are. In some sense, said H and V are incompatible. See below an illustration with 0 in white and 1 in black. The restrictions necessary lead to the faulty creation of an invalid column.



Remark 25. The choice of notation $X_{H,\mathcal{F}}$ can be slightly confusing at first, considering H is a one-dimensional SFT but $\mathcal{F}$ is a set of forbidden patterns. However, the most proper notation would first require to define $\mathcal{F}_H$ the forbidden words in H, then $\mathcal{F}'_H$ the extension of these into two-dimensional patterns (with no vertical restriction), and to denote $X_{\mathcal{F}'_H \cup \mathcal{F}}$ the resulting subshift. Similarly, $X_{H,V}$ would be more properly denoted as $X_{\mathcal{F}'_H \cup \mathcal{F}''_V}$, where $\mathcal{F}''_V$ is the extension of the forbidden words in $\mathcal{F}_V$ into two-dimensional patterns, but this time considered vertically, then extended horizontally with no restriction. Consequently, we decided to keep the notations simple and settle on what is used here.

4.3.2 Two new restricted Domino Problems

In Section 4.2.1, we saw that $DP(\mathbb{Z})$ was decidable, but $DP(\mathbb{Z}^2)$ was undecidable. In Chapter 5, and particularly in Section 5.1.2, we study an intermediary sort of Domino Problem, where horizontal constraints H are fixed beforehand and we look whether the emptiness of all the two-dimensional subsystems $X_{H,\mathcal{F}}$ is decidable.

Definition 4.14. Let $H \subset \mathcal{A}^{\mathbb{Z}}$ be an SFT. The *Domino Problem under horizontal constraints H* denoted as $DP_h(H)$ asks the decidability of the following set:

$$DP_h(H) = \{\langle \mathcal{F} \rangle \mid X_{H,\mathcal{F}} \text{ is a nonempty SFT}\}.$$

Remark 26. This Domino Problem is defined for a given H, and its decidability depends on such a H chosen beforehand.

The purpose of the previous definition is to determine the frontier between decidability ($DP(\mathbb{Z})$) and undecidability ($DP(\mathbb{Z}^2)$). Indeed, one can easily see that $DP(\mathbb{Z}) = DP_h(X_{\{00,11\}})$ where X is the full shift $\{0,1\}^{\mathbb{Z}}$; more generally it holds with any one-dimensional SFT – on an alphabet of cardinality at least two – made only of one periodic sequence and its translates. Similarly, $DP(\mathbb{Z}^2) = DP_h(X)$ or any one-dimensional full shift on an alphabet with cardinality at least two. Overall, “rigid” one-dimensional SFTs, with few configurations, seem to yield a decidable Domino Problem under horizontal constraints; conversely one-dimensional SFTs with a lot of freedom in their patterns seemingly make it undecidable. A complete result is obtained with Theorem 9.

In Chapter 6, and particularly in Section 6.2.2, we try to understand a close but distinct problem: when two one-dimensional SFTs are compatible to build a two-dimensional SFT. This question is notably reflected by the following adapted version of the Domino Problem:

Definition 4.15. Let $H \subset \mathcal{A}^{\mathbb{Z}}$ be an SFT. The *Domino Problem with Interplay depending on H* denoted as $DP_I(H)$ asks the decidability of the following set:

$$DP_I(H) := \{< V > \mid V \subset \mathcal{A}^{\mathbb{Z}} \text{ is an SFT and } X_{H,V} \neq \varnothing\}.$$

Remark 27. Just as for $DP_h(H)$, this problem is always defined for a given H (see Remark 32).

4.4 Root of a subshift

Given an SFT X, we want to know if it can simulate in some sense any two-dimensional subshift by adding local rules – as it is the question asked by DP_h and DP_I, and the topic of interest of Chapter 5 and Chapter 6. This may seem surprising, as one may ask how *adding* constraints to an SFT can lead to it somehow behaving like any other SFT. As it turns out, the simulation cannot be a conjugacy, because the entropy is preserved by conjugacy (see Proposition 4.10), or even a factor of the original subshift X.

Actually, there is as a "compensation" a change in scale in the simulation to obtain our result. Indeed, a more flexible way to reproduce a two-dimensional subshift Y's behavior is by looking at wider patches of configurations in X, called macro-tiles: depending on what these macro-tiles contain, they encode a number corresponding to an element of the alphabet of Y. As such, adding restrictions to X may allow these macro-tiles to be next to each other if and only if they respect the constraints of Y.

This idea of looking at macro-tiles is behind the notion of root of a subshift, as illustrated in Fig. 4.1:

Definition 4.16. The subshift $X \subset \mathcal{A}^{\mathbb{Z}^2}$ is a (m,n)*th root* of the subshift $Y \subset \mathcal{B}^{\mathbb{Z}^2}$ if there exist a clopen $Z \subset X$ with $\sigma^{(m,0)}(Z) = Z$ and $\sigma^{(0,n)}(Z) = Z$, and an homeomorphism $\varphi\colon Z \to Y$ such that:

- $\varphi(\sigma^{(k_1 m, k_2 n)}(x)) = \sigma^{(k_1,k_2)}(\varphi(x))$ for all $x \in Z$;
- $X = \bigsqcup_{0 \leq i < m, 0 \leq j < n} \sigma^{(i,j)}(Z)$.

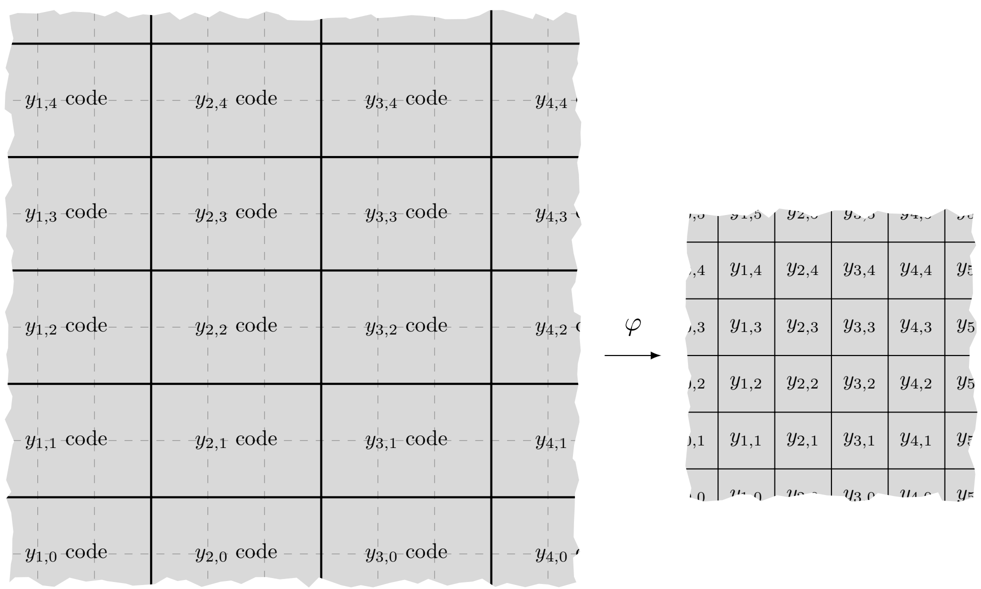


Figure 4.1: Illustration of the homeomorphism $\varphi\colon Z \to Y$ for a $(3,2)$ root, here on a configuration $z \in Z$ (left) that encodes a configuration $y \in Y$ (right) using 3×2 cells in X. The configurations in X are dashed-square shifts of the configurations in Z.

Chapter 5

Subsystems for Initial Horizontal Restrictions

A subsystem of a given dynamical system is merely a stable system included in the original one – that is, a topologically closed and action closed subspace of the original space. So is a subshift $X_{\mathcal{F}}$ subsystem of a given full shift; and so are all "sub-subshifts" $X_{\mathcal{F}\cup F'}$ of $X_{\mathcal{F}}$, obtained when adding forbidden patterns: they're subshifts that can be studied by themselves, or embedded into a bigger system.

In this chapter, we study the latter point of view: what are the accessible subsystems of a given subshift? What can we say about them in terms of Domino Problem (the focus of Section 5.1) and of accessible entropies (the focus of Section 5.2)? We restrain our study to horizontal starting constraints, as mentioned in the introduction of the present part, where we reach a complete answer for both questions.

5.1 Domino problem under horizontal constraints

5.1.1 Theorem of simulation under horizontal constraints

Proposition 5.1. *Let $H \subset \mathcal{A}^{\mathbb{Z}}$ be a one-dimensional SFT which is not made solely of eventually periodic configurations. For any two-dimensional SFT $Y \subset \mathcal{B}^{\mathbb{Z}^2}$, there exists a set of forbidden patterns $\mathcal{F}$ such that $X_{H,\mathcal{F}}$ is a (m,n)th root of Y for some $m, n \in \mathbb{Z}$.*

Proof. Let $Y \subset \mathcal{B}^{\mathbb{Z}^2}$ be a two-dimensional SFT. Up to renaming the symbols, suppose the alphabet $\mathcal{B}$ is made of letters $T_1, \dots, T_n$.

Suppose $H \subset \mathcal{A}^{\mathbb{Z}}$ is not solely made of eventually periodic configurations. Recall how H can be illustrated by a Rauzy graph, as detailed in Definition 1.28. Consider its Rauzy graph $\widetilde{\mathcal{G}}(H)$. We prove that there exists some vertex s of $\widetilde{\mathcal{G}}(H)$ so that there are at least two simple paths from s to itself (that is, paths that do not go through any vertex twice). The opposite would mean that each vertex has at most one simple path from itself to itself; notice how a vertex either has at least one simple such path, or no path at all, since from a non-simple path one can extract a simple one. Moreover, if some vertex additionally had a non-simple path from itself to itself that was not the iteration of its one simple path, notice how a vertex v repeated twice in that non-simple path would then have two distinct simple paths from itself to itself, which contradicts our current hypothesis. Therefore for any vertex with at least one simple path from itself from itself, there is no additional path that is not the iteration of its simple path. This causes $\widetilde{\mathcal{G}}(H)$ to be so that H has only eventually periodic configurations, which is forbidden.

Therefore, there exists some vertex s of $\widetilde{\mathcal{G}}(H)$ so that there are at least two simple paths from s to itself. Name those paths γ_1 and γ_2.

Define, for any k in $\{1, \ldots, n\}$,

$$U_k := \ell(\gamma_2\gamma_1\varphi_k\gamma_1\gamma_2)$$

where φ_k is a succession of n γ_1's, except the kth one, replaced by γ_2; and where $\ell(\gamma)$ designates the succession of labels of edges in a path γ, consequently being an element of $\mathcal{A}^*$ (the set of finite words made of elements of $\mathcal{A}$).

All of these words U_k have the same length, call it N, and we can juxtapose them as desired by construction of the Rauzy graph. Moreover, juxtaposing two of these words creates two consecutive γ_2's, allowing a clear segmentation of a word written with the U_k's into these basic units.

Now, consider the two-dimensional extension of H into $X_{H,\mathcal{F}}$ with $\mathcal{F}$ so that:

- we forbid anything horizontally that is not a succession of U_k's (possibly with different k's);
- we forbid any $\ell(\gamma_2\gamma_2)$ to be above something that is not another $\ell(\gamma_2\gamma_2)$ (this forces the U_k's to be vertically aligned);
- we forbid any pattern p of size $aN \times b$ made of U_k's that would be so that, if we consider the pattern q with T_l at position $(i,j), i \in \{0, \ldots, a-1\}, j \in \{0, \ldots, b-1\}$ if $p_{iN,j}$ belongs to U_l, we end up with a pattern q that is forbidden in Y.

This is a finite number of additional forbidden patterns, since Y is itself a SFT.

It is clear, considering the clopen Z of the configurations in $X_{H,\mathcal{F}}$ that have a U_k starting at $(0,0)$ exactly, and the homeomorphism $\varphi\colon Z \to Y$ that sends a U_k on a T_k, that $X_{H,\mathcal{F}}$ is a $(N,1)$th root of Y. See Fig. 5.1 for an illustration of the construction. □

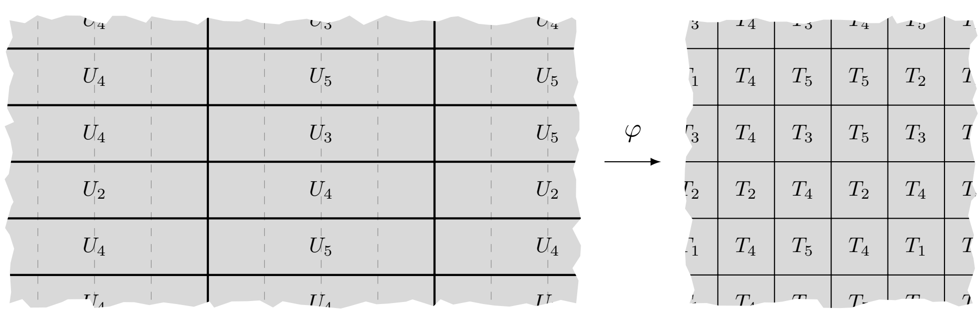


Figure 5.1: Illustration of the $(N,1)$ root built in the proof of Proposition 5.1, here with $N = 4$, through a homeomorphism $\varphi\colon Z \to Y$, here on a configuration $z \in Z$ (left) that encodes a configuration $y \in Y$ (right) using $N \times 1$ cells in X. The configurations in X are dashed-square shifts of the configurations in Z.

5.1.2 The Domino Problem under horizontal constraints

Theorem 9. *$DP_h(H)$ is decidable if and only if H contains only eventually periodic configurations.*

Remark 28. Since it is a SFT, it is equivalent to ask for H to have zero entropy. Indeed, as briefly seen in the proof of Proposition 5.1, containing only eventually periodic configurations is equivalent to restraining the number of simple paths in the Rauzy graph of H. The presence or absence of several simple paths from a vertex v to itself is ultimately – because it means that patterns of configurations in H can be, or not, replaced by other patterns of the same length – what is responsible for the exponential growth in the number of configurations in H, and so for a nonzero entropy. Through a similar line of reasoning, it is also equivalent to H containing countably many configurations.

Proof. If $H \subset \mathcal{A}^{\mathbb{Z}}$ has a configuration that is not eventually periodic, then for any two-dimensional SFT Y we can apply Proposition 5.1 and build a two-dimensional SFT $X_{H,\mathcal{F}}$ so that it is a root of Y. Using the definition of a root of a SFT, it is clear that the emptiness of $X_{H,\mathcal{F}}$ is equivalent to the emptiness of Y. Consequently, we can reduce $DP_h(H)$ to $DP(\mathbb{Z}^2)$, and $DP_h(H)$ is undecidable.

If $H \subset \mathcal{A}^{\mathbb{Z}}$ only has eventually periodic configurations, by compacity there is a configuration in a $X_{H,\mathcal{F}}$ if and only if there is one where each line is periodic; and so we can reason on actually periodic configurations of H only. Suppose these configurations' smallest common period (which exists as H is a SFT) is some integer $p > 0$. Now, take as input a finite set of forbidden patterns $\mathcal{F}$, all of them as a rectangle of size $pL \times M, L, M \in \mathbb{N}$ up to extending them.

If there is no rectangle of size $pL \times M(|\mathcal{A}|^{pLM} + 1)$ respecting the local rules of $X_{H,\mathcal{F}}$, then that SFT is empty. If there is at least one such rectangle, list all of these possible candidates. Consider R a candidate: either it can be horizontally juxtaposed with itself without containing a pattern of $\mathcal{F}$, and we keep it; or it cannot, and we delete it. If all candidates are deleted, then $X_{H,\mathcal{F}}$ is empty, because H forces a p-periodic repetition horizontally in any configuration, which happens to be incompatible with all candidate rectangles.

If at least one candidate R remains, then by the pigeonhole principle it contains at least twice the same rectangle R' of size $pL \times M$. To simplify the writing, we assume that the rectangle that repeats is the one with coordinates $[1, pL] \times [1, M]$ inside R where $[1, pL]$ and $[1, M]$ are intervals of integers, and that it can be found again with coordinates $[1, pL] \times [k, k + M - 1]$. Else, we simply truncate a part of R so that this becomes true.

Define $P := R|_{[1,pL]\times[1,k+M-1]}$. Since $\mathcal{F}$ has forbidden patterns of size $pL \times M$, and since R respects our local rules and begins and ends with R', P can be vertically juxtaposed with itself (overlapping on R'). Moreover, P can be horizontally glued with itself. The result tiles $\mathbb{Z}^2$ periodically while respecting the constraints of H and of $\mathcal{F}$, since any $pL \times M$ rectangle found in it is already present in the horizontal juxtaposition of P, which is valid by p-periodicity. Therefore, $X_{H,\mathcal{F}}$ is nonempty.

With this, we have an algorithm to decide the emptiness of $X_{H,\mathcal{F}}$ for any input $\mathcal{F}$. □

5.2 Characterization of the entropies under horizontal constraints

The purpose of this section is to characterize the entropies accessible to two-dimensional SFTs, as in [HM10], but under projective constraints. Formally, given H a one-dimensional SFT, we want to characterize the set

$$\{h(X_{H,\mathcal{F}}) \mid \mathcal{F} \text{ finite set of forbidden patterns}\}.$$

It is a set of positive reals by definition of the entropy; it is more precisely a subset of $[0, h(H)]$, as hinted at by [JKM07, Lemma 4.3 and Th. 4.1]. Indeed, by definition the highest available entropy in the set above is the one of $X_{H,\varnothing}$ – all of the other SFTs considered have a set of forbidden patterns of which this one's is a subset. The entropy of that two-dimensional SFT, as shown by a straightforward computation[1], is also the one of the one-dimensional SFT H.

A result by Desai [Des06], when applied to the subsystems of $X_{H,\varnothing}$, states the following:

Proposition 5.2 (Th. 3.3 from [Des06])**.** $\{h(X_{H,\mathcal{F}}) \mid \mathcal{F}$ *finite set of forbidden patterns*$\}$ *is dense in* $[0, h(H)]$.

Moreover, the computability obstruction of [HM10] reminded in Theorem 8 implies that that the previous set is a subset of the Π_1^0-computable real numbers (also named *right-recursively enumerable* or r.r.e. numbers). Therefore, it is natural to ask if all Π_1^0-computable numbers of $[0, h(H)]$ can be obtained. We have the following positive result proved in Section 5.2.3.

Theorem 11. *Let* $H \subset \mathcal{A}^{\mathbb{Z}}$ *be a SFT. We have*

$$\{h(X_{H,\mathcal{F}}) \mid \mathcal{F} \text{ finite set of forbiden patterns}\} = [0, h(H)] \cap \Pi_1^0$$

where Π_1^0 *is the set of right-recursively enumerable real numbers.*

5.2.1 Kolmogorov complexity and number of tiles

Given r a Π_1^0-computable real, denote $K(r)$ its *Kolmogorov complexity*. It is the minimal number of states needed for a Turing Machine to enumerate a list of rationals which approach r from above. Consider the algorithm from [HM10, Alg. 7.3], defined for a given r.r.e. real h, that takes as input a sequence $(x_N)_{N\in\mathbb{Z}} \in \{0,1\}^{\mathbb{Z}}$, and makes its frequency of 1's – on specific indices – approach r from above. The associated Turing Machine is built from the one that approaches h from above so that its number of states is of the form $cK(h)$, $c > 0$ not depending on h. The authors in [HM10] turn that Turing Machine into a Wang tile set. Since the size of that tile set depends linearly on the number of states of the Turing Machine, we obtain the following:

Theorem 10 (From [HM10], Alg. 7.3)**.** *There exists* $C > 0$ *such that for any* $h \in \Pi_1^0$*, there exists a two-dimensional SFT* X *such that* $h(X) = h$*, describable by a set of at most* $CK(h)$ *Wang tiles.*

5.2.2 Technical lemmas on entropy

Lemma 5.3. *For all* $\alpha, \beta \in \mathbb{N}$*, one has:*

$$\lim_{n\to+\infty} \frac{\log_2(N_X(\alpha n, \beta n))}{\alpha\beta n^2} = h(X).$$

Proof. We have the inequality

$$N_X(\alpha n, \beta n) \leq (N_X(n,n))^{\alpha\beta}$$

[1] Notice how the definition of entropy is used for a two-dimensional SFT on one hand, and for a one-dimensional one on the other hand, though the resulting real number is the same.

due to the fact that globally admissible patterns – patterns that belong to a valid configuration – of size $\alpha n \times \beta n$ are themselves made of a number $\alpha\beta$ of globally admissible patterns of size $n \times n$.

We also have:

$$(N_X(\alpha\beta n, \alpha\beta n)) \leq N_X(\alpha n, \beta n)^{\alpha\beta}$$

with the same reasoning.

Finally, we have

$$\lim_n \frac{\log_2(N_X(\alpha\beta n, \alpha\beta n))}{\alpha^2\beta^2 n^2} = h(X)$$

because $(\frac{\log_2(N_X(\alpha\beta n,\alpha\beta n))}{\alpha^2\beta^2 n^2})$ is a subsequence of the converging sequence $(\frac{\log_2(N_X(n,n))}{n^2})$.

Then applying $\lim_n \dfrac{\log_2(.)}{\alpha\beta n^2}$ to

$$(N_X(\alpha\beta n, \alpha\beta n))^{\frac{1}{\alpha\beta}} \leq N_X(\alpha n, \beta n) \leq (N_X(n,n))^{\alpha\beta}$$

we obtain what we want. □

Proposition 5.4. *Let X be a two-dimensional subshift which is a (m,n)th root of the subshift Y. Then*

$$h(Y) = mn\, h(X)$$

Proof. Call $Z \subset X$ the clopen homeomorphic to Y. In what follows, we write $\mathcal{L}_C^{(k_1,k_2)}(a,b)$ the set of $[k_1, k_1+a-1] \times [k_2, k_2+b-1]$ patterns that appear in configurations of a clopen C; and $N_C^{(k_1,k_2)}(a,b)$ its cardinal. Note that $\mathcal{L}_X^{(0,0)}(a,b) = \mathcal{L}_X(a,b)$ since X is shift-invariant; and same for Y.

Consider $\varphi : Z \to Y$ the homeomorphism such that $\varphi(\sigma^{(k_1 m, k_2 n)}(x)) = \sigma^{(k_1,k_2)}\varphi(x)$ for all $x \in Z$ and $(k_1,k_2) \in \mathbb{Z}^2$. By the same proof as the Curtis-Hedlund-Lyndon theorem, there exist $r \in \mathbb{N}_0$ and a local map $\overline{\varphi}$ applied on patterns of support $[-r,r]^2$ such that for any $x \in Z$, $\varphi(x)_{(k_1,k_2)} = \overline{\varphi}(x_{(k_1 m, k_2 n)+[-r,r]^2})$ – that is, φ can be considered as the application of a local map uniformly on patterns of configurations of Z. Thus $|\mathcal{L}_Y^{(0,0)}(k,k)| \leq |\mathcal{L}_Z^{(-r,-r)}(mk+2r, nk+2r)|$. Furthermore, since a pattern of support $[-r, mk+r-1] \times [-r, nk+r-1]$ can be decomposed into a pattern of support $[0, mk-1] \times [0, nk-1]$ and its border, we deduce that $|\mathcal{L}_Y^{(0,0)}(k,k)| \leq |\mathcal{A}_Z|^{2rk(m+n)} |\mathcal{L}_Z^{(0,0)}(mk,nk)|$.

In the same way, there exist $r' \in \mathbb{N}_0$ and a local map $\overline{\varphi^{-1}}$ applied on patterns of support $[-r',r']^2$ such that $\varphi^{-1}(y)_{(k_1 m, k_2 n)+[0,m-1]\times[0,n-1]} = \overline{\varphi^{-1}}(y_{(k_1,k_2)+[-r',r']^2})$. Thus $|\mathcal{L}_Z^{(0,0)}(mk,nk)| \leq |\mathcal{L}_Y^{(-r',-r')}(k+2r', k+2r')|$

We have the same bounds for any $\sigma^{(i,j)}(Z)$ with $i,j \in \mathbb{Z}$. Moreover, since we have that $X = \bigsqcup_{0\leq i<m, 0\leq j<n} \sigma^{(i,j)}(Z)$, we obtain $N_X(mk,nk) = \sum_{0\leq i<m,0\leq j<n} N^{(0,0)}_{\sigma^{(i,j)}(Z)}(mk,nk)$. We deduce the following inequalities

$$\frac{mn}{|\mathcal{A}_Z|^{2rk(m+n)}} N_Y(k,k) \leq N_X(mk,nk) \leq mn N_Y(k+2r', k+2r')$$

We apply $\frac{\log_2(.)}{k^2}$ to these inequalities and by using Lemma 5.3, we get:

$$h(Y) = mnh(X).$$

□

Lemma 5.5. *Let H be a transitive one-dimensional SFT of positive entropy and $h < h(H)$. There exist words $u, w_1, w_2 \in \mathcal{L}_H$ and $\alpha \in \mathbb{N}$ such that:*

- *$\alpha > M$ the biggest size of forbidden patterns of H;*
- *$|u| = |w_1| = |w_2| = \alpha$;*
- *u, w_1 and w_2 are cycles from a vertex of the Rauzy graph of H to the same vertex;*
- *$uww_1 \in \mathcal{L}_H$ for all $w \in \{w_1, w_2\}^*$;*
- *u appears as a subword of any word in $u\{w_1, w_2\}^* w_1$ at the very beginning only;*
- *$h(H_u) > h$ where H_u is the SFT included in H where the word u does not appear.*

Moreover, the Rauzy graph of H_u is still transitive.

Proof. According to [Lin89, Th.3], for any integer n that is big enough, any n-long word u is so that, if we denote H_u the SFT where u is added to the forbidden patterns of H, we get

$$h(H) \geq h(H_u) > h$$

because $h(H_u)$ is below yet sufficiently close to $h(H)$ for n big enough. As a consequence, the last property we have to find is verified as soon as we choose α big enough for the three words we want.

Now, consider the Rauzy graph of H, call it $\widetilde{\mathcal{G}}(H)$. Fix some vertex s of $\widetilde{\mathcal{G}}(H)$. Since its entropy is positive and it is transitive, for any vertex s of $\widetilde{\mathcal{G}}(H)$ there are at least two simple paths from s to itself (notably, that do not pass through s at any other moment). Name those paths γ_1 and γ_2.

Define

$$\begin{aligned} u &:= \ell(\gamma_2\gamma_1\gamma_2\gamma_1\gamma_1\gamma_1\gamma_2\gamma_1{}^k\gamma_2) \\ w_1 &:= \ell(\gamma_2\gamma_1\gamma_1\gamma_2\gamma_1\gamma_1\gamma_2\gamma_1{}^k\gamma_2) \\ w_2 &:= \ell(\gamma_2\gamma_1\gamma_1\gamma_1\gamma_2\gamma_1\gamma_2\gamma_1{}^k\gamma_2) \end{aligned}$$

where $\ell(\gamma)$ designates the succession of labels of edges in γ, consequently being an element of $\mathcal{A}^*$. Choose $k > 0$ big enough so that u satisfies what we desire on $h(H_u)$. These three words have the same length and we can juxtapose them as desired by construction of the Rauzy graph of H. Moreover, juxtaposing any of those words creates two consecutive γ_2, allowing a clear segmentation of a word written with u, w_1 and w_2 into these basic units. Notably, u appears as a subword of any word in $u\{w_1, w_2\}^* w_1$ at the very beginning only.

Moreover, H_u is still transitive. Indeed, let v_1 and v_2 be two elements of $\mathcal{L}_{H_u}$; we can suppose they are bigger than u up to an extension of v_1 to its left and v_2 to its right.

v_1 can be seen as a path in $\widetilde{\mathcal{G}}(H)$ and extended to $v_1 w$ so that the corresponding path in $\widetilde{\mathcal{G}}(H)$ reaches s, by transitivity of that graph, with the shortest possible w. It is possible that $v_1 w \notin \mathcal{L}_{H_u}$, meaning it contains u as a subword. But then u cannot be a subword of w, since the path in $\widetilde{\mathcal{G}}(H)$ corresponding to u is a succession of cycles from s to s; and u cannot be a subword of v_1, since $v_1 \in \mathcal{L}(H_u)$. Therefore u begins in v_1 and ends in w. Since w corresponds by definition to the shortest path to s possible in $\widetilde{\mathcal{G}}(H)$, it means most of u is in v_1 and only a fraction of the last $\ell(\gamma_2)$ is in w. Then change w for a new w' as short as possible that breaks the completion of u: this is doable because $v_1 \in \mathcal{L}_{H_u}$, so it is extendable to the right. Then add the shortest possible w'' that goes to s in $\widetilde{\mathcal{G}}(H)$: u is too large to appear elsewhere in $v_1 w' w''$. Rename $w'w''$ as w.

In all cases, we found $v_1 w \in \mathcal{L}_{H_u}$ that reaches s when considered in $\widetilde{\mathcal{G}}(H)$. Do the same so that some $xv_2 \in \mathcal{L}_{H_u}$ starts from s when considered in $\widetilde{\mathcal{G}}(H)$.

Now, $v_1 w x v_2 \in \mathcal{L}_H$ since v_1 and v_2 are large enough so that the wx part of this word has no effect on its extension to a biinfinite word in H_u. Furthermore, $v_1 w x v_2$ does not contain u, since the latter must start from s and end at s, and this word has been built so that no u follows when s appears. Since v_1 and v_2 are also large enough to be extended respectively to the left and to the right without a u, we can conclude that $v_1 w x v_2 \in \mathcal{L}_{H_u}$. Hence $\widetilde{\mathcal{G}}(H_u)$ is transitive. □

The following lemma is colloquially called the Coin Problem or Frobenius Coin Problem. We produce a simple, self-contained proof here. For a more straightforward argument, one can use Schur's theorem from combinatorics that ensures the existence of a rank N as in the lemma.

Lemma 5.6 (Frobenius Coin Problem)**.** *Let $c_1, \ldots, c_l$ be positive integers. Let $m = GCD(c_1, \ldots, c_l)$.*

There is a rank $N \in \mathbb{N}$ such that for all $n \geq N$, nm can be expressed as some $\sum_{i=1}^{l} k_i c_i$ with $k_i \in \mathbb{N}$.

Proof. Using Bézout's identity, we can write $m = \sum_{i=1}^{l} z_i c_i$ for integers $z_i \in \mathbb{Z}$.

Let $c = \sum_{i=1}^{l} c_i$. m divides c. Any multiple nm of m can be written as $nm = qc + rm$ where q, r are integers with $0 \leq r < \frac{c}{m}$. Notably, $nm = \sum_{i=1}^{l} (q + r z_i) c_i$ using the previous equalities with m and c.

If $n \geq 2\frac{c^2}{m^2} \max_i |z_i|$, then $nm \geq \frac{c^2}{m} \max_i |z_i| + \frac{c^2}{m} \max_i |z_i|$. Since $qc + \frac{c^2}{m} \max_i |z_i| > qc + rc \max_i |z_i| \geq qc + rm = nm$, we deduce that $q > \frac{c}{m} \max_i |z_i| > r \max_i |z_i|$.

Therefore, if $n \geq 2\frac{c^2}{m^2} \max_i |z_i|$, then for all i, $q + r z_i > 0$. As a consequence, $nm = \sum_{i=1}^{l} k_i c_i$ with $k_i = q + r z_i > 0$. □

Lemma 5.7. *Let H be a transitive one-dimensional SFT of positive entropy. Let u and w_1 be the words defined as in Lemma 5.5, with $\alpha = |u| = |w_1|$.*

Let

$$\widetilde{N_H}(n) = Card(\{v \mid |v| = n, u \not\sqsubset w_1 v u_{[0,\alpha-2]} \text{ and } w_1 v u \in \mathcal{L}_H\}).$$

Then

$$\frac{\log_2(\widetilde{N_H}(n\alpha))}{n\alpha} \xrightarrow[n\to\infty]{} h(H_u).$$

Proof. Let s be the vertex of the Rauzy graph of H that begins and ends the paths corresponding to u and w_1. We will similarly name s its label, a word in ${\mathcal{A}_H}^*$.

Let m be the GCD of the lengths of all cycles c_i from s to s that do not pass through s at any other moment, in the Rauzy graph of H. From Lemma 5.6, we deduce that there is some $N \in \mathbb{N}$ such that for any $n \geq N$, there is a path of length nm from s to s in the Rauzy graph of H, as a concatenation of all cycles c_i with a positive number of times k_i for each of them. Since any order of concatenation works, one can notably build a cycle from s to s of length nm that does not contain u (by concatenating all occurrences of path γ_2, as defined in the proof of Lemma 5.5 in a row).

Let d be the diameter of the Rauzy graph of H_u. Let $n \geq N + \frac{2d+|u|}{\alpha}$. Let $v \in \mathcal{L}_{H_u}$ so that $|v| = n\alpha - 2d - N\alpha$ (notably $|v| \geq |u|$). Since the graph $\widetilde{\mathcal{G}}(H_u)$ is transitive, there are two words v' and v'' of size smaller than d so that $v' v v'' \in \mathcal{L}_{H_u}$, with the path corresponding to v' beginning with some vertex $s*$ when seen as a path in $\widetilde{\mathcal{G}}(H_u)$, and the path corresponding to v'' ending with some vertex $*s$ – where $s*$ is the label of a vertex that has s as a prefix, and $*s$ the label of a vertex that has s as a suffix. This word can also be seen as a path in the Rauzy graph of H; there, that path is a cycle from s to s; hence $|v'vv''| = km$ for some $k \in \mathbb{N}$ since m divides the length of all cycles from s to s in $\widetilde{\mathcal{G}}(H)$.

A notable consequence of this, using the length of v and the fact that m divides α (since w_1 corresponds to a cycle from s to s in H) is that $2d - |v'| - |v''| > 0$ is a multiple of m.

Let v''' be a word of length $2d + N\alpha - |v'| - |v''|$ that is a cycle from s to s in H. This is doable by doing a correct concatenation of cycles from s to s to build v''', because $2d + N\alpha - |v'| - |v''|$ is a multiple of m which is greater than $N\alpha$, hence it is greater than Nm, so we can apply Lemma 5.6. We can also ask that $v'vv''v''' \in \mathcal{L}_{H_u}$: since the path corresponding to v in $\widetilde{\mathcal{G}}(H)$ does not contain s as a vertex except at its extremities, and up to a reduction of the length of v'' so that it is the same, the only risk is that v''' itself would contain u. However, there are several cycles from s to s in $\widetilde{\mathcal{G}}(H)$ (since the corresponding SFT has positive entropy). Even if v''' were to use the two that we label γ_1 and γ_2 in the proof of Lemma 5.5 to build u, we could merely order them with all γ_1 paths first, then all γ_2 paths – since we only care about the length of the resulting word – to build v'''. Thus, we forbid any appearance of u in it.

It remains to prove that $w_1v'vv''v'''u_{[0,\alpha-2]} \in \mathcal{L}_{H_u}$. Consider that word as a path in $\widetilde{\mathcal{G}}(H)$: as u starts and ends at vertex s, the only subwords/subpaths of $w_1v'vv''v'''u_{[0,\alpha-2]}$ we have to be wary of are of length α and between vertices s and s. Considering v corresponds to a path that does not cross the vertex s aside from its extremities, no word u can begin or end in v. Up to taking a shorter v' (resp. v''), we can also make its corresponding path start (resp. end) with a vertex $s*$ (resp. $*s$) in $\widetilde{\mathcal{G}}_{H_u}$, without crossing any other vertex of the sort. Since $v'vv''v''' \in \mathcal{L}_{H_u}$, the only remaining risk is of a subword u that would start in v''' and end in $u_{[0,\alpha-2]}$. As mentioned before, if we construct v''' correctly, that risk can be avoided too.

We have $|v'vv''v'''| = n\alpha$, $w_1v'vv''v'''u_{[0,\alpha-2]} \in \mathcal{L}_{H_u}$ and $w_1v'vv''v'''u \in \mathcal{L}_H$.

We deduce that

$$\widetilde{N_H}(n\alpha) \geq N_{H_u}(n\alpha - 2d - N\alpha).$$

Since we also have $N_{H_u}(n\alpha) \geq \widetilde{N_H}(n\alpha)$, taking the logarithm and dividing by $n\alpha$, we obtain

$$\frac{\log_2(\widetilde{N_H}(n\alpha))}{n\alpha} \underset{n\to\infty}{\longrightarrow} h(H_u).$$

□

5.2.3 Main result

Theorem 11. *Let $H \subset \mathcal{A}^{\mathbb{Z}}$ be a SFT. We have*

$$\{h(X_{H,\mathcal{F}}) \mid \mathcal{F} \textit{ finite set of forbiden patterns}\} = [0, h(H)] \cap \Pi_1^0$$

where Π_1^0 is the set of right-recursively enumerable real numbers.

Proof. Let $h \leq h(H), h \in \Pi_1^0$. We assume that $h(H) > h > 0$; if not, a trivial two-dimensional SFT satisfies the Theorem. It is possible to assume that H is transitive: indeed, in one-dimensional SFTs the entropy is due to only one connected component of the Rauzy graph (see [LM95]). Furthermore, since we assumed $h(H) > 0$, that connected component is not a simple cycle, which are of entropy 0. Hence we can additionally assume that the Rauzy graph of H is not a simple cycle.

Since H is a transitive SFT with positive entropy, thanks to Lemma 5.5 there exists a word $u \in \mathcal{L}_H$ of size $\alpha = |u|$ larger than the order of H and two different words $w_1, w_2 \in L(H)$ such that for $|w_1| = |w_2| = \alpha$ such that $uww_1 \in \mathcal{L}_H$ for all $w \in \{w_1, w_2\}^*$. Moreover $h(H_u) > h$ where H_u is the subshift included in H where the word u does not appear.

Denote
$$\widetilde{N_H}(n) = Card(\{v \mid |v| = n, u \not\sqsubset w_1 v u_{[0,\alpha-2]} \text{ and } w_1 v u \in \mathcal{L}_H\}).$$

According to Lemma 5.7, one has

$$\frac{\log_2(\widetilde{N_H}(n\alpha))}{n\alpha} \xrightarrow[n\to\infty]{} h(H_u).$$

Consider an integer $t \geq 2$ such that $h(H_u) > (1 + \frac{1}{t})h$. Moreover, we ask for t to be big enough so that for any $n \geq t$,
$$\frac{\log_2(\widetilde{N_H}(n\alpha))}{n\alpha} > (1 + \frac{1}{t})h$$
which is possible since the left term converges to $h(H_u)$.

Now consider K the sum of the Kolmogorov complexities of the following different elements: h, t, α, the algorithms that compute the addition, the multiplication, the logarithm, the algorithm that on input n returns $\widetilde{N_H}(n)$ and the algorithm that on input (a, b, c) trio of integers returns the first integer r such that $\frac{a}{b+r-1} > c \geq \frac{a}{b+r}$.

Let $R = \lceil \log_2(3CK) \rceil$ and consider $q = Rt$. One has

$$\frac{\log_2(\widetilde{N_H}(q\alpha))}{(R+q)\alpha} = \frac{q}{q+R}\frac{\log_2(\widetilde{N_H}(q\alpha))}{q\alpha} > \frac{t}{t+1}(1 + \frac{1}{t})h = h.$$

Therefore, there exists $r > R$ such that

$$\frac{\log_2(\widetilde{N_H}(q\alpha))}{(q+r-1)\alpha} > h \geq \frac{\log_2(\widetilde{N_H}(q\alpha))}{(q+r)\alpha}$$

due to the fact that it is a sequence decreasing to 0 as r increases.

Consider $h' = (q+r)\alpha \left(h - \frac{\log_2(\widetilde{N_H}(q\alpha))}{(q+r)\alpha} \right) > 0$. The Kolmogorov complexity of h' is less than $3K$, because K contains the complexity of doing each of the operations used in h' except computing q, and computing q requires to compute R which requires to compute K which has Kolmogorov complexity at most K. Assembling all of this, we obtain a complexity of at most $3K$.

Consequently, using Theorem 10, there exists a constant $C > 0$ and a Wang tile set T_W with at most $3CK$ tiles such that the associated SFT W has entropy $h(W) = h'$.

Now, consider the two-dimensional subshift $X \subset \mathcal{A}^{\mathbb{Z}^2}$ with the following local rules:

- every line satisfies the conditions of H, thus $\pi_{\vec{e_1}}(X) \subset H$;
- if the word u appears horizontally starting at position (i, j), it also appears at positions $(i, j+1)$ and $(i, j-1)$;

- the word u appears with the horizontal period $(q+r)\alpha$ and it cannot appear elsewhere on a given line;
- the word u is followed by a word of $\{w_1, w_2\}^R.w_1^{r-R-1}$;
- the tiles coded in binary by the words of $\{w_1, w_2\}^R$ after u satisfy horizontal and vertical constraints imposed by the Wang tile set T_W.

The window of size $q\alpha$ that remains between two words u and is not filled by the previous constraints has the only restriction of respecting the horizontal conditions of H and of not containing u. There are horizontal lines that respect all of the horizontal conditions in X, because:

- $u\{w_1, w_2\}^* w_1^* \subset \mathcal{L}_H$ by the use of Lemma 5.5;
- $\widetilde{N_H}(q\alpha) \neq 0$ by the use of Lemma 5.7;
- forcing u to appear nowhere else than with a $(q+r)\alpha$ period is possible because of Lemma 5.5: u appears as a subword of any word in $u\{w_1, w_2\}^* w_1$ at the very beginning only, and it appears nowhere else due to the very definition of $\widetilde{N_H}(q\alpha)$.

The only vertical restriction that is added between these horizontal lines is only the alignment of the u's between two lines where this word appears; so there are configurations in X overall. See Fig. 5.2.

Let $n = (q+r)\alpha$. One has

$$\widetilde{N_H}(q\alpha)^{k(k-1)} N_W(k-1, k) \leq N_X(kn, k)$$

because in any $kn \times k$ window in X there are at least $(k-1) \times k$ complete horizontal segments starting with a word u, that encode a $k-1 \times k$ pattern of W binary with $\{w_1, w_2\}^R$, and each of them also contains $q\alpha$ additional characters that link w_1 and u and that do not contain u.

One also has

$$N_X(kn, k) \leq \widetilde{N_H}(q\alpha)^{k(k+1)} N_W(k+1, k)$$

because similarly, in any $kn \times k$ window in X there are less than $(k+1) \times k$ complete horizontal segments starting with a word u. We obtain the following inequality:

$$\frac{(k-1)\log_2(\widetilde{N_H}(q\alpha))}{kn} + \frac{\log_2(N_W(k-1,k))}{nk^2} \leq \frac{\log_2(N_X(kn,k))}{nk^2} \leq \frac{(k+1)\log_2(\widetilde{N_H}(q\alpha))}{kn} + \frac{\log_2(N_W(k+1,k))}{nk^2}$$

Thus, taking the limit when $k \to \infty$, one obtains

$$h(X) = \frac{\log_2(\widetilde{N_H}(q\alpha))}{(q+r)\alpha} + \frac{h(W)}{(q+r)\alpha} = h.$$

□

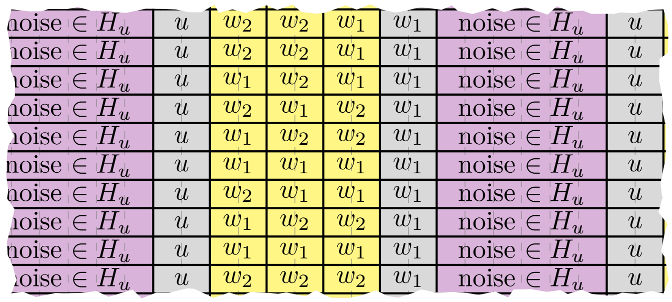


Figure 5.2: Illustration (not to scale) of the proof, representing a two-dimensional subshift X with projective subaction in H and any given entropy $h \in [0, h(H)] \cap \Pi_1^0$. Most of h is given by the noise zone in purple, which are words chosen from $\{v \mid |v| = n, u \not\sqsubset v \text{ and } w_1 v u \in \mathcal{L}_H\}$, with a big entropy "diluted" in the larger-scale construction. The rest of it is due to the simulation of a carefully-chosen Wang shift in the highlighted zone.

5.2.4 Some consequences

A direct consequence of Theorem 11 – up to extending the construction to higher dimensions, which is straightforward – is the characterization of all the entropies in any dimension $d > 2$.

Corollary 5.8. *Let $\mathcal{A}$ be a finite alphabet. For any number $h \in \Pi_1^0$ such that $0 \leq h \leq \log_2(|\mathcal{A}|)$, there exists a SFT on $\mathcal{A}^{\mathbb{Z}^d}$ of entropy h.*

Though the above result looks like the main theorem from [HM10], it is actually more precise. Hochman and Meyerovitch's construction realizes these entropies with what is seemingly an arbitrarily big alphabet; here we show that any fixed alphabet size allows any entropy it can (with a $\log_2(|\mathcal{A}|)$ bound, the alphabet's size naturally limiting the attainable entropies due to Proposition 4.10).

For what follows, we introduce the notion of projective subaction of a two-dimensional subshift $X \subset \mathcal{A}^{\mathbb{Z}^2}$: the set of all one-dimensional configurations that appear as patterns of two-dimensional configurations of X, following a given integer vector $\vec{v}$. It is denoted as $\pi_{\vec{v}}(X)$. Most notably, $\pi_{\vec{e_1}}(X) = \{y \in \mathcal{A}^{\mathbb{Z}} \mid \exists x \in X, \exists k \in \mathbb{Z}, \forall i \in \mathbb{Z}, x_{i,k} = y_i\}$ is the horizontal projective subaction of X. It is a one-dimensional subshift.

Given a one-dimensional *effective subshift* H, that is a subshift with a list of forbidden patterns enumerable by a Turing Machine, it is known that there exists a two-dimensional sofic subshift which has H as projective subaction [AS13]. However, in the theorem of simulation of [AS13], but that sofic subshift has zero entropy. A natural question is if it is possible to force a specific entropy h with $h \leq h(H)$ – to aim for any Π_1^0-computable number of $[0, h(H)]$. The next corollary is a partial result for effective subshifts containing a SFT.

Remark 29. This question is related to the conjecture that a one-dimensional subshift H is sofic if and only if the two-dimensional subshift $H^* = \{x \in \mathcal{A}^{\mathbb{Z}^2} : \text{ for all } i \in \mathbb{Z}, x_{\mathbb{Z}\times\{i\}} \in H\}$ is sofic. We remark that the entropy of H^* is $h(H)$, obtained with completely independent rows; thus allowing entropy in the realization is a way of giving some independence between rows.

Corollary 5.9. *Let $H \subset \mathcal{A}^{\mathbb{Z}}$ be an effective subshift with $H' \subset H$ a SFT. Let $0 \leq h \leq h(H'), h \in \Pi_1^0$. Then there exists a two-dimensional sofic subshift X such that $h(X) = h$ and $\pi_{\vec{e_1}}(X) = H$.*

Proof. Let $h \leq h(H'), h \in \Pi_1^0$.

Let X' be the two-dimensional SFT obtained from H' using Theorem 11, which is so that $h(X') = h$ and $\pi_{\vec{e_1}}(X') \subset H'$.

Let Y_H be the SFT with H as horizontal rules and no added vertical rule.

Let Z be a SFT over $\{0, 0', 1\}^{\mathbb{Z}^2}$ so that two horizontally successive elements must be the same, so that only 0 and 1 can be above 0, only $0'$ can be above 1, and only $0'$ can be above $0'$. The configurations of Z are made of at most a single line of 1s with $0'$s above and 0s below.

We define $Y = Z \times Y_H \times Y'$ which is a SFT by product, and finally X which is the projection $\pi(Y)$ with

$$\pi(y)_{i,j} = \begin{cases} (y_2)_{i,j} & \text{if } (y_1)_{i,j} = 1 \\ (y_3)_{i,j} & \text{if } (y_1)_{i,j} \neq 1 \end{cases}$$

where y_1, y_2 and y_3 are the projections of a configuration $y \in Y$ on the three SFTs of the product.

In the end, a configuration in X has at most one line that can be any configuration of H; and all its other lines are from X', ensuring $h(X) = h(X') = h$. Furthermore, since $\pi_{\vec{e_1}}(X') \subset H' \subset H$, we obtain $\pi_{\vec{e_1}}(X) = H$. □

Chapter 6

Interplay Between Horizontal and Vertical Conditions

A different way of looking at one-dimensional constraints in a two-dimensional setting, as mentioned in Section 4.3, is to try to understand if a pair of constraints are compatible. Some are chosen as horizontal conditions and the others as vertical conditions: can the resulting combined subshift $X_{H,V}$ be anything we want?

6.1 Theorem of simulation under interplay

Apart from what is done in Chapter 5, another way of looking at one-dimensional constraints in a two-dimensional setting is to try to understand if a pair of constraints are compatible. Some are chosen beforehand as horizontal conditions H: now, can we find vertical conditions V so that the resulting combined subshift $X_{H,V}$ exhibits the same possible behavior as any subshift we want?

For a few nearest-neighbor horizontal constraints, it is not that hard to realize that, whichever vertical constraints we match them with, the combined subshift will necessarily contain periodic configurations, and therefore not every SFT on $\mathbb{Z}^2$ will be imitable in behavior. These constraints are said to respect condition D, see Section 6.1.2, which is the union of three smaller easy-to-understand conditions.

For any other kind of horizontal constraint, we prove in a disjunctive fashion that we can simulate, in some sense, any two-dimensional subshift Y. However, since the entropy of $X_{H,V}$ is bounded by $h(H)$, the simulation cannot be a conjugacy; the correct notion is the one of root of a subshift, defined in Section 4.4.

This section is devoted to proving the following, for a condition D to be defined in Section 6.1.2.

Theorem 12. *Let $H \subset \mathcal{A}^{\mathbb{Z}}$ be a one-dimensional* nearest-neighbor *SFT whose Rauzy graph does not satisfy condition D. For any two-dimensional SFT $Y \subset \mathcal{B}^{\mathbb{Z}^2}$, there exists a one-dimensional SFT $V_Y \subset \mathcal{A}^{\mathbb{Z}}$ such that X_{H,V_Y} is a (m,n)th root of Y for some $m, n \in \mathbb{N}^2$. Furthermore, m, n and V_Y can be computed algorithmically.*

6.1.1 Core idea

We have a one-dimensional nearest-neighbor SFT $H \subset \mathcal{A}^{\mathbb{Z}}$ ("horizontal") that does not verify condition D, and we fix a Wang shift W with a set of N tiles $\tau = \{\tau_1, ..., \tau_N\}$.

The idea is to introduce a well-chosen one-dimensional SFT $V \subset \mathcal{A}^{\mathbb{Z}}$ ("vertical") depending on W so that $X_{H,V}$ encodes the full shift on N elements. Then, we refine V by adding conditions on forbidden patterns, thus encoding exactly the configurations in W. Such a construction is done with the use of two main parts, that we will obtain by some carefully chosen forbidden patterns in V.

First, there are parts of synchronization, also called *sync parts*, that give some rigidity to our tilings. They precise where the actual coding parts can be, which letters of the alphabet can be used and where in these coding parts, and they ensure that you cannot glue patches together in an unexpected way. They are the frame of our construction. Second comes the filling: the *coding parts*. A given coding part simply codes a number between 1 and N, possibly several times.

In Fig. 6.1a (our first rough attempt to encode a full shift on an alphabet of size N), we suppose that our sync parts properly maintain this global structure. We notice that it offers an interesting opportunity to transmit information vertically. Since our coding parts are exactly aligned, once we have encoded the full shift over an alphabet of size N, it will suffice to add vertical conditions to our V to precise whether a coding part can be above another one.

However, horizontally Fig. 6.1a overlooks two problems:

- Since we must respect the horizontal conditions given by H, we cannot put any coding part next to any other one if we do not put some kind of buffer between the two;

- Even with this, we have no control on the horizontal transfer of information. The idea is to transmit this horizontal information vertically, since we can add vertical constraints.

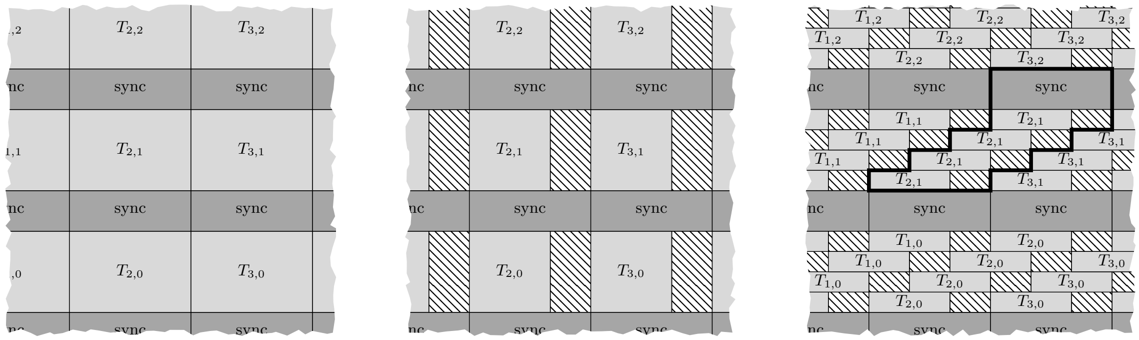
We can fix the first problem by setting a buffer (see Fig. 6.1b) between two coding parts, a portion of column that contains no coding and which can be next to any coding part. Of course, we must ensure that this buffer cannot be anywhere in a configuration but obediently remains between two coding parts. The sync parts will be designed to handle this.

However, this does not solve our need for horizontal transmission of information. Hence a new idea: altering our coding parts so that they transmit information diagonally. We put several consecutive lines of them, shifted little by little, as illustrated in Fig. 6.1c. That way, we can encode horizontal forbidden patterns vertically, because we can see vertically which coding part is on the right of the one we are considering. For instance, by looking vertically we can know that the encoding of $T_{1,1}$ is next to $T_{2,1}$ and above $T_{1,0}$, and thus restrict the content of these codings.

In what follows, we will build Fig. 6.1c in details, although some technicalities will be needed to preserve the integrity of our sync parts and to ensure that the coding of a tile of W is well transmitted. This construction will indeed encode the full shift over τ, the tile set of W. Then, it can easily be refined by adding vertical rules so that the local rules of W are ensured. Consequently, our newly built $X_{H,V}$ will properly simulate all configurations of W, allowing us to perform the rest of the proof of Theorem 12.

6.1.2 The condition D

We recall the notion of strongly connected component, abridged as SCC, mentioned below Property 1.30, as a connected component of a directed graph.



(a) Basic depiction of sync parts and coding parts that represent tiles of W (not to scale; actually unrealizable).

(b) Same construction adding "buffers" between codings of tiles to be realizable.

(c) Improved construction: we encode vertically the horizontal restrictions between tiles of W. A tile of W (here $T_{2,1}$) is represented in bold.

Figure 6.1: Steps of the core idea to reach the generic construction. Not to scale: the sync part will be much bigger than the code part.

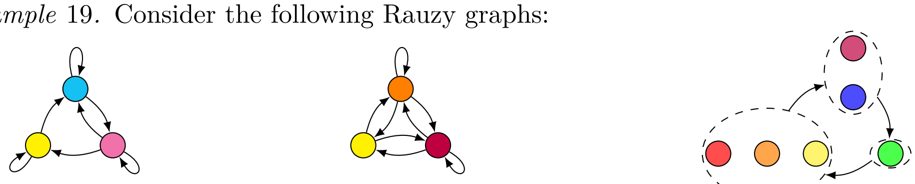
Definition 6.1. We say that an oriented graph $\mathcal{G} = (\mathcal{V}, \vec{E})$ verifies condition D (for "Decidable") if all its SCCs have a type in common among the following list. A SCC S can be of none, one or several of these types:

- for all vertices $v \in S$, we have $(v, v) \in \vec{E}$: we say that S is of *reflexive type*;
- for all vertices $v \neq w \in S$ such that $(v, w) \in \vec{E}$, we have $(w, v) \in \vec{E}$: we say that S is of *symmetric type*;
- there exists $p \in \mathbb{N}$ so that $S = \bigsqcup_{i=0}^{p-1} V_i$ with, for any $v \in V_i$, we have $[(v, w) \in \vec{E} \Leftrightarrow w \in V_{i+1}]$ with $i + 1$ meant modulo p: we say that S is of *state-split cycle type*.

Remark 30. The term state-split is used in reference to a notion introduced in [LM95]: a state-split cycle is a cycle where some vertices have been split.

Note that $S = \{v\}$ a single vertex with a loop is also of symmetric type. Similarly, a single vertex is of state-split cycle type with a partition with one unique class V_0.

Example 19. Consider the following Rauzy graphs:



$\widetilde{\mathcal{G}}(H_1)$ $\widetilde{\mathcal{G}}(H_2)$ $\widetilde{\mathcal{G}}(H_3)$

where edges between dotted sets of vertices in the third graph represent that all vertices from the first set have edges leading to all vertices of the second set.

These three graphs respect condition D, being respectively of reflexive, symmetric and state-split cycle type.

6.1.3 Generic construction

In this section, we describe a set of properties on a directed graph, forming a condition called *condition C* that has stronger requirements than condition D. Condition C allows a generic construction of the proof of Theorem 12 for the associated one-dimensional, nearest-neighbor SFT.

In all that follows, *we denote elements of cycles with an index that is written modulo the length of the corresponding cycle.* We need the following defintions before describing condition C.

Definition 6.2. Let C^1 and C^2 be two distinct cycles in an oriented graph $\mathcal{G}$, with elements denoted respectively c^1_i and c^2_j. Let $M := LCM(|C^1|, |C^2|)$. Let C be any cycle in that graph, with elements denoted c_i.

We say that the cycles C^1 and C^2 contain a *good pair* if there is a pair (i, j) and an integer $1 < l < M-1$ such that $c^1_i \neq c^2_j, c^1_{i+1} \neq c^2_{j+1}, \ldots, c^1_{i+l} \neq c^2_{j+l}$ and $c^1_{i+(l+1)} = c^2_{j+(l+1)}, \ldots, c^1_{i+(M-1)} = c^2_{j+(M-1)}$. All pairs $(i+p, j+p), p \in \{0, \ldots, M-1\}$ are said to be *in the orbit of a good pair.*

We say that a cycle C *contains a uniform shortcut* if there exists a $k \in \{0, 2, 3..., |C|-1\}$ (any value except 1) such that for any $c_i \in C, (c_i, c_{i+k}) \in \vec{E}$.

We say that there is a *cross-bridge* between C^1 and C^2 if there are $i \in \{0, ..., |C^1|-1\}$ and $j \in \{0, ..., |C^2|-1\}$ with $c^1_i \neq c^2_j$ and $c^1_{i+1} \neq c^2_{j+1}$ such that $(c^1_i, c^2_{j+1}) \in \vec{E}$ and $(c^2_j, c^1_{i+1}) \in \vec{E}$.

Finally, an *attractive vertex* for a set of vertices S is any vertex v so that for all $s \in S$, $(s, v) \in \vec{E}$. A *repulsive vertex* for S is defined similarly, with $(v, s) \in \vec{E}$ instead.

These definitions are illustrated in Fig. 6.2.

Definition 6.3. Let $H \subset \mathcal{A}^{\mathbb{Z}}$ be a one-dimensional nearest-neighbor SFT. We say that H *verifies condition C* if $\widetilde{\mathcal{G}}(H) = (\mathcal{V}, \vec{E})$ contains two cycles C^1 and C^2, of elements denoted respectively c^1_i and c^2_j, with the following properties:

(i) $|C^1| \geq 3$;

(ii) C^1 and C^2 contain a good pair;

(iii) There is no uniform shortcut neither in C^1 nor in C^2;

(iv) There is no cross-bridge between C^1 and C^2;

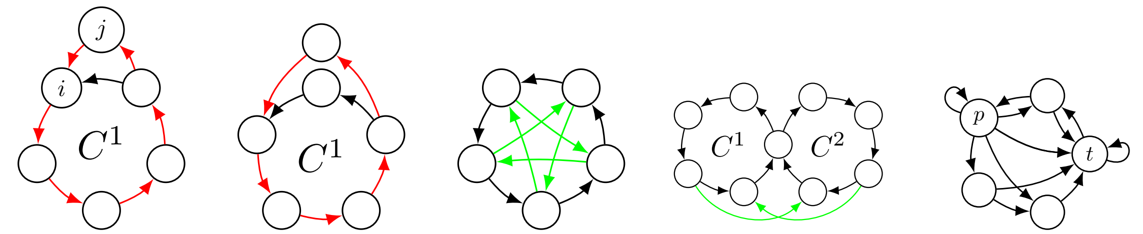
(v) C^1 contains no vertex that is both attractive and repulsive vertex in $C^1 \cup C^2$ (seen as a set).

Some vertices can be *repeated*; that is, these cycles can go several times through the same vertex as long as they verify the aforementioned properties.

Proposition 6.4. *If $\widetilde{\mathcal{G}}(H)$ verifies condition C, and W is a Wang shift, then there exists an SFT $V_W \subset \mathcal{A}^{\mathbb{Z}}$ that can be algorithmically derived from W, such that X_{H,V_W} is a root of W.*

The rest of the subsection is devoted to proving this result.

Let H with $\widetilde{\mathcal{G}}(H)$ verifying condition C. We focus on encoding as a root, in the correct $X_{H,V}$, a full shift on an alphabet τ of cardinality N. Then, the possibility to add vertical rules will allow us to encode any Wang shift W using this alphabet, that is, to simulate the configurations of W as a root of $X_{H,V}$.



(a) C^1 and C^2 (in red) have a good pair (i, j).

(b) C^1 and C^2 (in red) have no good pair.

(c) A cycle with uniform shortcuts of distance 3 (in green).

(d) A cross-bridge (in green) between cycles C^1 and C^2.

(e) An attractive vertex t and a repulsive vertex p.

Figure 6.2: Cases of compliance or not with elements of Condition C.

For the rest of the construction, we name $M := LCM(|C^1|, |C^2|)$ and $K := 2|C^1| + |C^2| + 3$. We suppose that $N \geq 2$. Indeed, the case of encoding a monotile Wang shift is easy: consider only the cycle C^1 in $\widetilde{\mathcal{G}}(H)$, that may have extra edges from one element to another, but no uniform shortcut. Build V the vertical SFT from the graph $\widetilde{\mathcal{G}}(H)'$ obtained by removing any of those extra edges, keeping only a simple cycle – the same as C^1. Then $X_{H,V}$ contains only the translate of one configuration, that cycles through the elements of C^1 in the correct order, both horizontally and vertically. This is a root of a monotile Wang shift.

We refer to Fig. 6.3a in the description that follows. We use the term *slice* as a truncation of a column: it is a part of width 1 and of finite height. We use the following more specific denominations for the various scales of our construction:

- A *macro-slice* is a slice of height KMN. Any column of $X_{H,V}$ will merely be made of a succession of some specific macro-slices called ordered macro-slices (see below).
- A *meso-slice* is a slice of height MN; meso-slices are assembled into macro-slices.
- A *micro-slice* is a slice of height N. This subdivision is used inside specific meso-slices called code meso-slices (see below).

Although any scale of slice could denote any truncation of column of the right size, we focus on specific slices that are meaningful because of what they contain, so that we can assemble them precisely. They are:

- An (i, j) k-coding micro-slice is a micro-slice composed of $N - 1$ symbols c_i^1 and one symbol c_j^2 at position k. It encodes the kth tile of alphabet τ, unless $c_i^1 = c_j^2$: in that case, it is called a *buffer* and encodes nothing. We can write "(i, j) coding micro-slice" when we do not want to specify which tile is encoded.
- An (i_0, j_0) (k, l)-code meso-slice is a meso-slice made of M vertically successive coding micro-slices. The one on top is an (i_0, j_0) k-coding micro-slice. We add the following restrictions:
  - an (i, j) k-coding micro-slice must be vertically followed by a $(i + 1, j + 1)$ k-coding micro-slice, unless $c_i^1 \neq c_j^2$ but $c_{i+1}^1 = c_{j+1}^2$, that is if the new micro-slice is a buffer but the previous one is not;

  – if $c_i^1 = c_j^2$ but $c_{i+1}^1 \neq c_{j+1}^2$, then the buffer must be followed by a $(i+1, j+1)$ l-coding micro-slice.

  We can write "(i_0, j_0) code meso-slice" when we do not want to specify which tiles are encoded.

- An i-border meso-slice is made of $\frac{M}{|C^1|}N$ times the vertical repetition of all the elements of the cycle C^1, starting with c_i^1.
- A c_i^1 meso-slice is made of MN times the vertical repetition of element c_i^1, denoted $(c_i^1)^{MN}$ in Fig. 6.3a. The same definition holds for a c_j^2 meso-slice.
- The succession of a c_i^1 meso-slice, then a c_{i+1}^1 meso-slice, ..., then a c_{i-1}^1 meso-slice is called a i C^1-slice. It is of height $MN|C^1|$. Similarly, we define a j C^2-slice (of height $MN|C^2|$).
- Finally, a (i,j)-ordered (k,l)-coding macro-slice is the succession of a i-border meso-slice, a i C^1-slice, a second i C^1-slice, a j C^2-slice, a i-border meso-slice, and finally a (i,j) (k,l)-code meso-slice. We can write "(i,j)-ordered macro-slice" when we do not want to specify which tiles are encoded.

Remark 31. The (i_0, j_0) (k,l)-code meso-slice is well-defined because since C^1 and C^2 contain a good pair, a code meso-slice is a vertical succession of coding micro-slide and of buffers, with either only one vertical succession of coding micro-slice of one vertical succession of buffers (possibly both). Therefore, there can be at most one change, from k to l, in the tiles encoded; k is then called the *main-coded tile*, and l the *side-coded tile*.

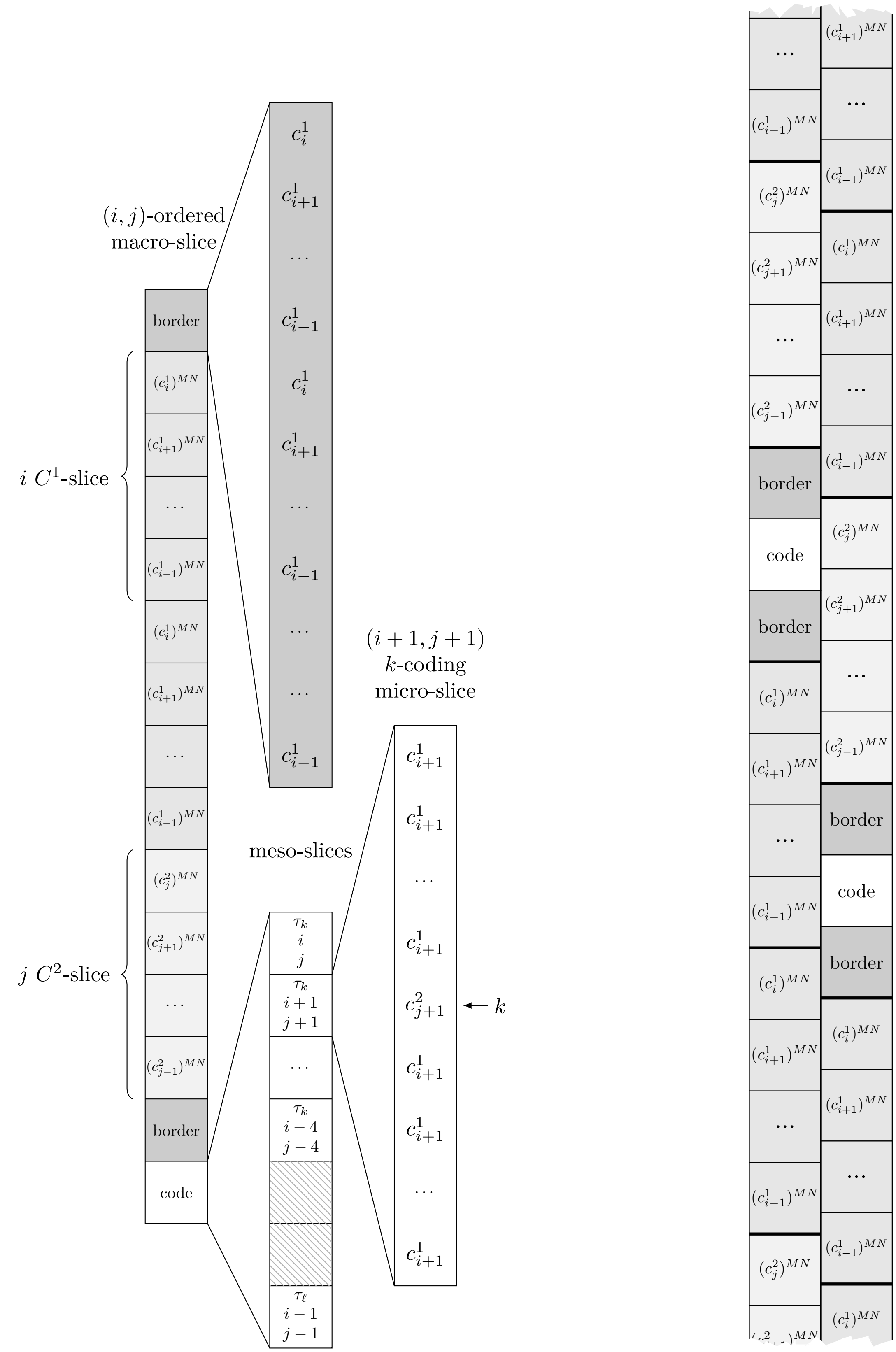
Note that if $c_{i_0}^1 \neq c_{j_0}^2$ but $c_{i_0-1}^1 = c_{j_0-1}^2$, then the meso-slice contains no side-coded tile – its last coding micro-slice is the last buffer – so l is actually irrelevant. Conversely, if $c_{i_0}^1 = c_{j_0}^2$, then the meso-slice contains no main-coded tile, so k is actually irrelevant.

Now, the patterns we authorize in V are the (i_0+p, j_0+p)-ordered macro-slices with a good pair (i_0, j_0) and $p \in \{0, \ldots, M-1\}$, and all patterns that allow the vertical juxtaposition of two (i,j)-ordered macro-slices, using the same i and j, but possibly different code meso-slices. We prove below that this is enough for our resulting $X_{H,V}$ to simulate a full shift on τ.

We say that two legally adjacent columns are *aligned* if they are subdivided into ordered macro-slices exactly on the same lines. We say that two adjacent and aligned columns are *synchronized* if any (i,j)-ordered macro-slice of the first one is followed by a $(i+1, j+1)$-ordered macro-slice in the second one.

Proposition 6.5. *In this construction, two legally adjacent columns are aligned up to a vertical translation of size at most $2|C^1|-1$ of one of the columns.*

Proof. If two columns, call them K_1 and K_2, can be legally juxtaposed such that they are not aligned even when vertically shifted by $2|C^1|-1$ elements, it means that one of the border meso-slices of K_1 has at least $2|C^1|$ vertically consecutive elements that are horizontally followed by something that is not a border meso-slice in K_2 (see Fig. 6.3b). Since $2|C^1| < MN$ which is the length of a meso-slice, at least $|C^1|$ successive elements among the ones of the border meso-slice in K_1 are horizontally followed by elements that are part of the same meso-slice in K_2. If this is a code meso-slice, simply consider the other border meso-slice of K_1 (the first you can find, above or below, before repeating the pattern cyclically): that one must be in contact with a c_i^1 or c_j^2 meso-slice instead. Either way, we obtain that a border meso-slice has at least $|C^1|$ successive elements that are horizontally followed



(a) Columns allowed, for (i,j) in the orbit of a good pair. Here $c^1_{i-3} = c^2_{j-3}$ and $c^1_{i-2} = c^2_{j-2}$, forming buffers in the code meso-tile, represented as hatched squares.

(b) Faulty alignment of adjacent columns.

Figure 6.3: The generic construction.

by some t meso-slice made of a single element t. Hence if we suppose that juxtaposing K_1 and K_2 this way is legal, it means that in H all the elements of C^1 lead to t, i.e t is an attractive vertex. Either this is forbidden, or the "reverse" reasoning where we focus on the borders of K_2 proves that there is also an element p used in a $C^i, i \in \{1,2\}$ slice of K_1 that leads to every element of C^1; that is, C^1 has a repulsive vertex in $C^1 \cup C^2$. Condition C forbids any graph that had both, hence we reach a contradiction. We obtain the proposition we announced. □

Proposition 6.6. *In this construction, two legally adjacent columns are always aligned and synchronized.*

Proof. Proposition 6.5 states that two adjacent columns K_1 and K_2 are always, in some sense, approximately aligned up to a vertical translation of size at most $2|C^1|-1$. If the two columns are slightly shifted still, then any meso-slice of the C^1 slices of K_1 (such a meso-slice consists only of the repetition of some c_i^1) is horizontally followed by two different meso-slices in K_2. Being different, at least one of them is not c_{i+1}^1 but some $c_{i+k}^1, k \in \{2, ..., |C^1|\}$. This is true with the same k for all values of i, notably because an ordered macro-slice contains two successive C^1 slices, so all meso-slices representing c_i^1 are repeated twice – there is no problem with the meso-slices at the extremities. We obtain something that contradicts our assumption that C^1 has no uniform shortcut. Hence there is no vertical shift at all between two consecutive columns. Thus our construction ensures that two adjacent columns are always aligned.

It is easy to see that a meso-slice made only of c_i^1 in column K_1 is horizontally followed, because the columns are aligned, by a meso-slice made only of c_{i+k}^1 in column K_2, for some $k \in \{0, \ldots, |C^1|-1\}$. This k is once again independent of the i because inside a macro-slice, meso-slices respect the order of cycle C^1. But because C^1 has no uniform shortcut, we must have $k=1$. The reasoning is the same for the C^2 slice, and we use the fact that C^2 has no shortcut either. Hence our columns are synchronized. □

With these properties, we have ensured that our structure is rigid: our ordered macro-slices are aligned and synchronized. The last fact to check is the transmission of information, represented by the following proposition:

Proposition 6.7. *In this construction, an (i,j)-ordered (k,l)-coding macro-slice is horizontally followed by an $(i+1,j+1)$-ordered (k,l)-coding macro-slice, except in two situations:*

- *if $c_i^1 = c_j^2$, we can have a (k,l)-coding macro-slice followed by a (k',l)-coding macro-slice;*
- *if $c_i^1 \neq c_j^2$ but $c_{i-1}^1 = c_{j-1}^2$, we can have a (k,l)-coding macro-slice followed by a (k,l')-coding macro-slice.*

Proof. The exceptions are due to an earlier remark: if $c_i^1 = c_j^2$, then the code meso-slice contains no main-coded tile, so its value of k is actually irrelevant. Similarly, if $c_i^1 \neq c_j^2$ but $c_{i-1}^1 = c_{j-1}^2$, the code meso-slice contains no side-coded tile, so its value of l is irrelevant.

For the rest of the proof, it is already clear from Proposition 6.6 that an (i,j)-ordered macro-slice is horizontally followed by an $(i+1,j+1)$-ordered macro-slice; the only part left to study is the coding part.

Now, consider two horizontally adjacent coding micro-slices. By synchronicity, one is an (i,j) micro-slice, and the other an $(i+1,j+1)$ micro-slice. Since $\widetilde{\mathcal{G}}(H)$ verifies condition C, and particularly has no cross-bridge, we cannot have both an edge $(c_i^1, c_{j+1}^2) \in \vec{E}$ and an edge $(c_j^2, c_{i+1}^1) \in \vec{E}$, except

Figure 6.4: A Rauzy graph and several associated code meso-slices for $|\tau| = 3$. Here are horizontally successively encoded τ_3, τ_1, τ_2 and τ_2, the number being indicated by the location of the line of c's. One can check that τ_1 can be located left to τ_2 in the encoding, by using vertical constraints only, as depicted by the bold rectangle with rounded corners.

if one of the two micro-slices is a buffer. Therefore, either one of them is a buffer, or they both are p-coding for the same p.

This is enough to prove the proposition. □

In the end, we proved that if we were able to find two cycles C^1 and C^2 complying with condition C, they would be enough to build the construction we desire: the root of a full shift on N elements. Indeed, take Z the clopen made of all the configurations with, at position $(0,0)$, the bottom of an (i,j)-ordered macro-slice with $c^1_{i-1} = c^2_{j-1}$ but $c^1_i \neq c^2_j$. Suppose that macro-slice is (k,l)-coding (with an irrelevant l). Map it, along with the $M-1$ that follow horizontally (hence, map a $M \times KMN$ rectangle) on $y_{(0,0)} = \tau_k$. By mapping the whole configuration similarly, Z is homeomorphic to Y with the required properties: X_{H,V_Y} is therefore a (M, KMN)th root of $\tau^{\mathbb{Z}^2}$.

Then, to encode only the configurations that are valid in W a Wang shift, we forbid the following additional vertical patterns:

- the code meso-slices that would contain both τ_k as its main-coded tile and τ_l as its side-coded tile, if τ_k cannot be horizontally followed by τ_l in W;
- and the vertical succession of two ordered macro-slices that would contain code meso-slices with two main-coded tiles that cannot be vertically successive in W.

With this, we proved Proposition 6.4: the set of vertical conditions obtained, that define a one-dimensional SFT V_W, is such that X_{H,V_W} is a (M, KMN)th root of W.

6.1.4 Summary of the generic construction for one strongly connected component

We suppose that $H \subset \mathcal{A}^{\mathbb{Z}}$ is a one-dimensional nearest-neighbor SFT such that its Rauzy graph $\widetilde{\mathcal{G}}(H) = (V, \vec{E})$ does not verify condition D and is made of only one SCC.

In all that follows, we say that a vertex u *has a loop* if $(u,u) \in \vec{E}$ – it is *loopless* otherwise. We name *unidirectional* any edge (u,v) of $\vec{E}$ so that there is no edge (v,u) in $\vec{E}$ – it is *bidirectional* if both are in $\vec{E}$. Note that $\widetilde{\mathcal{G}}(H)$, since it does not verify condition D, contains at least one loopless vertex, and one unidirectional edge.

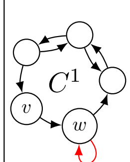
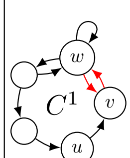
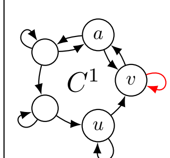
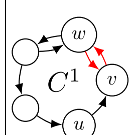
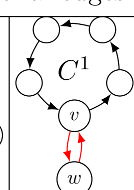
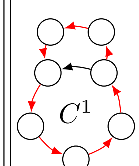
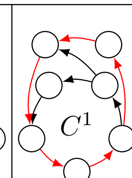
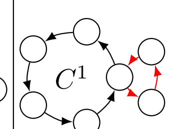
The notion of *cycle* is intended as an ordered tuple of vertices representing the order of traversal. Except when *explicitly* mentioned otherwise, all paths and cycles manipulated in the following subsections are assumed to be *simple*, that is, they do not self-intersect by going through the same vertex twice (except for a path from a vertex to itself, that is, a cycle). For ease of reading, the fact that we manipulate simple paths and cycles until further notice is occasionally reminded to the reader throughout the following subsection.

The reasoning that follows is based on defining two cycles C^1 and C^2, and trying to fit condition C from Definition 6.3 as much as possible. We name C^1's vertices c^1_i and C^2's vertices c^2_j, with $i \in \{0, \ldots, |C^1|-1\}$ and $j \in \{0, \ldots, |C^2|-1\}$.

The idea of the proof of Theorem 12 is to classify the possible graphs into various cases. In each case, one has a standard procedure to find convenient C^1 and C^2 inside any graph to perform the generic construction from Section 6.1.3. For some specific cases, we will not meet condition C even though H does not verify condition D. However we prove even when one or two items of condition C fail that the propositions from Section 6.1.3 that are Proposition 6.5 and Proposition 6.6 still hold nonetheless, and the consequent reasoning too.

The division into cases is presented in a disjunctive fashion, and broadly reproduced in Table 6.1.

Table 6.1: Table of the main cases, each of them illustrated with an example (the C^1 on which we perform the generic construction is the main cycle indicated; the C^2 is in red).

<table>
<tr><td colspan="3">Loops</td><td colspan="5">No loop</td></tr>
<tr><td></td><td></td><td></td><td colspan="2">Bidirectional edges</td><td colspan="3">No bidirectional edge</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Case 1.1</td><td>Case 1.2</td><td>Case 1.3</td><td>Case 2.1</td><td>Case 2.2</td><td>Case 3.1</td><td>Case 3.2</td><td>Case 3.3</td></tr>
</table>

Is there a loop on a vertex?

- If YES, this is *Case 1* (Section 6.1.5); is there a unidirectional edge between one vertex with a loop and one loopless vertex?
  - If YES: this is *Case 1.1*.
    - ∗ either all items of condition C hold for the cycles we focus on, and we perform the generic construction;

        * or there exist both attractive and repulsive vertices for the cycle C^1 defined – but Proposition 6.5 still holds because one vertex has in-degree exactly 1.

    – If NO: are all unidirectional edges between two loopless vertices?

        * If YES: this is *Case 1.2*.
            · first option, in trying to find convenient cycles, we are able to reduce to a subgraph similar to the ones of Case 2.2, and apply that;
            · second option, we find convenient cycles on which all items of condition C hold, and we perform the generic construction;
            · third option, we reduce to a length 3 case as in Section 6.1.8.
        * If NO: this is *Case 1.3*; all unidirectional edges are between two vertices with loops.
            · first option, all items of condition C hold for the cycles we focus on, and we perform the generic construction;
            · second option, cross-bridges exist between C^1 and C^2. We prove that in this specific case, only one kind of cross-bridge can happen that actually endangers the generic construction. Either it does not, and we still perform the generic construction; or it does, but then C^1 is of length 3, which is solved in Section 6.1.8.
            · last option, an attractive and a repulsive vertex both exist for C^1; then we prove C^1 has at most five vertices, a specific case which is solved in Section 6.1.8.

- If NO, is there at least one bidirectional edge?

    – If YES, this is *Case 2* (Section 6.1.6); can we find a simple cycle of length at least 3 containing at least one bidirectional edge?

        * If YES: this is *Case 2.1*. In this case we easily find cycles that verify condition C and we perform the generic construction.
        * if NO: this is *Case 2.2*. We can find cycles so that all items of condition C hold for them, and we perform the generic construction.

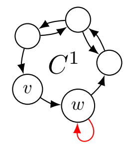
    – If NO, this is *Case 3* (Section 6.1.7); considering the smallest cycle C one can find, is there a path between two different vertices of C?

        * If YES, can one find such a path so that it does not intersect C elsewhere and is of a length different from the distance between the vertices it links when measured in C?
            · If YES: this is *Case 3.1*. Either all items of condition C hold for the cycles we focus on – up to swapping which one is used as C^1 and which one is used as C^2 (possible because both have at least three vertices) – and we perform the generic construction; or there are uniform shortcuts; then we carefully choose new cycles to focus on and prove that we can perform the generic construction on these new cycles, some way or other.
            · If NO: this is *Case 3.2*. Then condition C holds for the cycles we focus on – up to swapping which one is used as C^1 and which one is used as C^2 here too – and we perform the generic construction.
        * If NO, this is *Case 3.3*. Then once again, for the cycles we focus on, we show that they verify condition C and we can perform the generic construction.

6.1.5 Case 1

We suppose that $\widetilde{\mathcal{G}}(H)$ contains a loop.

Case 1.1: we can find a unidirectional edge so that the first vertex is loopless and the second has a loop (or the opposite, for which the construction is similar and omitted).

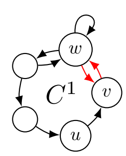


Take the shortest possible cycle containing such an edge, which exists since $\widetilde{\mathcal{G}}(H)$ is strongly connected. Call v the loopless vertex and w the vertex with a loop. Naming that cycle C^1 with $c_0^1 = w$, and setting $C^2 = \{w\}$, we have to check that they fulfill the conditions of Section 6.1.3.

(v, w) is unidirectional and v is loopless. Note that no edge can go from w to any vertex that is not w or c_1^1, else we could find a shorter cycle with the same characteristics. Similarly, v has in-degree 1. Hence:

- $|C^1| \geq 3$ since $(w, v) \notin \vec{E}$;
- C^1 and C^2 contain a good pair, that is (c_1^1, w).
- C^2 has no uniform shortcut because it is made of one single vertex. C^1 has no uniform shortcut because of what precedes about w and because v is loopless;
- If there was a cross-bridge between C^1 and C^2, it would mean there are two edges (c_i^1, w) and $(w, c_{i+1}^1) \in \vec{E}$ with $c_i^1 \neq w \neq c_{i+1}^1$, which is also impossible because of what precedes about w;
- Here, there can actually be attractive and repulsive vertices for C^1, which endangers Proposition 6.5. However, in this specific case, the only vertex that has an edge going to v is the previous one in the cycle C^1, call it $u := c_{-2}^1$. As such, in any column, the v meso-slice must be next to the u meso-slice of the previous column since no other block of u of size MN can be found in said previous column. Hence two consecutive columns are always aligned and we can make the generic construction work with no restriction on attractive and repulsive vertices. As such, Proposition 6.5 still holds, albeit for reasons different from the ones in Section 6.1.3: we fully exploit the fact that v is of in-degree 1.

Case 1.2: all unidirectional edges are between two loopless vertices.



Since $\widetilde{\mathcal{G}}(H)$ is in Case 1 and Subcase 1.2, it contains loops and its unidirectional edges are necessarily between two loopless vertices. Hence it contains bidirectional edges. Moreover, it cannot contain only bidirectional edges, else it would verify condition D. Hence $\widetilde{\mathcal{G}}(H)$ contains both unidirectional and bidirectional edges.

We name u, v, w three successive vertices in the graph so that $(u, v), (v, w)$ and $(w, v) \in \vec{E}$, and $(v, u) \notin \vec{E}$ (so u and v, additionally, have no loop). Two situations are possible: either there is a path from w to u that does not go through v, and we obtain by taking the shortest of these a cycle containing both a unidirectional and a bidirectional edge; or there is not. In that second case, we consider a path from v to u:

- either it contains no bidirectional edge, and we focus on the subgraph made of one cycle with unidirectional edges and the bidirectional edge (v, w): this is treated just as Case 2.2 from Section 6.1.6 (with the extra of w having a loop, but this does not change the reasoning).
- or the path from v to u contains a bidirectional edge; then concatenated to (u, v) it forms a cycle with both a unidirectional and a bidirectional edge.

Iteratively reducing the cycle obtained to the shortest one possible, we either end up in a situation that can be reduced to Case 2.2, or to a cycle similar to the figure above: with a unidirectional edge, followed by a bidirectional edge, and containing no shorter cycle that would fit.

Naming that cycle C^1 with $c^1_0 = v$, and defining C^2 as the cycle containing only v and w, we have to check that they fulfill the conditions of the generic construction. Note that w cannot lead to any vertex except c^1_2, v, and possibly w; else we could find a cycle shorter than C^1 that has the same properties. Note, also, that $v \neq c^1_2$, else we would reduce to Case 2.2. Hence:

- $|C^1| \geq 3$;
- C^1 and C^2 have a good pair: (c^1_2, v) is one.
- C^2 has no uniform shortcut because it is made of only two vertices and v has no loop. C^1 has no uniform shortcut of length 0 because u and v are loopless, and no uniform shortcut of length -1 because $(v, u) \notin \vec{E}$. With what precedes about w, there is no uniform shortcut at all;
- If there was a cross-bridge, we could have two cases:
  - First is (c^1_i, v) and $(w, c^1_{i+1}) \in \vec{E}$ with $c^1_i \neq w$ and $c^1_{i+1} \neq v$. Since v has no loop, we deduce $c^1_i \neq v$, hence $c^1_{i+1} \neq w$. Also consider that $c^1_{i+1} = c^1_2$ would imply $c^1_i = w$, which is impossible. Therefore, with what we said on w, that kind of cross-bridge cannot happen.
  - Second is (c^1_i, w) and $(v, c^1_{i+1}) \in \vec{E}$ with $c^1_i \neq v$ and $c^1_{i+1} \neq w$. Since v has no loop and $(v, u) \notin \vec{E}$, we also deduce $v \neq c^1_{i+1}$ and $u \neq c^1_{i+1}$. Then we can define a shorter cycle that is $C^{1\prime} = (v, c^1_{i+1}, c^1_{i+2}, ..., u)$. Since $(v, u) \notin \vec{E}$, $C^{1\prime}$ has length at least 3, contains at least one unidirectional edge, and is strictly shorter than C^1. Since C^1 is a minimal cycle having these properties and containing a bidirectional edge, $C^{1\prime}$ must contain no bidirectional edge. Then $C^{1\prime}$ is a cycle of unidirectional edges, such that a length two cycle (made of v and w) shares one vertex in common with it. This is an iterative reduction that should have already been performed to build C^1 and C^2, therefore it cannot happen here.
- If there is an attractive vertex for C^1 located in C^2, then it is in particular in C^1 since $C^2 \subset C^1$. Since they have no loop, u and v can't be attractive or repulsive. If any other vertex than w or the following vertex, call it x, was attractive, then it would allow for a direct

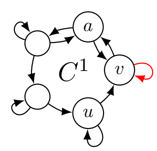
edge from w to that vertex, and so a fitting cycle strictly shorter than C^1 would exist, which is impossible. But if w (resp. x) was attractive, in particular $(u, w) \in \vec{E}$ (resp. $(u, x) \in \vec{E}$). Since w (resp. x) would have a loop because it also attracts itself, in this Case 1.2 we couldn't have a unidirectional edge (u, w) (resp. (u, x)): necessarily $(w, u) \in \vec{E}$ (resp. $(x, u) \in \vec{E}$). The only possibility not to cause a contradiction with the minimality of C^1 is that C^1 is already of length 3 (resp. 4).

The length 3 case is treated in Section 6.1.8. The length 4 case with (u, v, w, x) and x attractive is actually impossible, because (u, v, x) reduces to a length 3 case with the correct properties, but C^1 was supposed to be minimal.

With all this, we have reduced all situations faced in Case 1.2 to one of the following:

- a specific case of Section 6.1.8;
- or a reasoning similar to case 2.2 from Section 6.1.6;
- or an impossibility;
- or something that actually follows condition C (Definition 6.3).

Case 1.3: all unidirectional edges are between two vertices with loops.



Since $\widetilde{\mathcal{G}}(H)$ does not verify condition D, we can find a cycle with at least one loopless vertex and one unidirectional edge, that, in this specific case, may go through the same vertices twice since it is defined, strictly speaking, as the concatenation of one path from the unidirectional edge to the loopless vertex, and one path back, the two of them being able to intersect.

Exceptionally we will manipulate here a non-simple, self-intersecting cycle: define C^1 as the smallest cycle with the concatenation of the two aforementioned paths. Call u and v the two successive vertices of the unidirectional edge, that is $(u, v) \in \vec{E}$ but $(v, u) \notin \vec{E}$. Note that u and v have a loop since we are in Case 1.3. Moreover, call a the loopless vertex that was also used to build C^1. Finally, set $a = c_0^1, u = c_{i_0}^1, v = c_{i_0+1}^1$.

Setting $C^2 = \{v\}$, we have to check that they fulfill the conditions of the generic construction:

- $|C^1| \geq 3$;
- $(c_{i_0+2}^1, v)$ is a good pair since C^1 passes through v only once in the cycle by construction.
- C^2 is made of only one vertex, hence it has no uniform shortcut. C^1 has no uniform shortcut of length 0 since a has no loop. It has no uniform shortcut of length -1 because $(v, u) \notin \vec{E}$. We study a hypothetical shortcut that would allow $(a, c_j^1) \in \vec{E}$: notice that $(c_j^1, a) \in \vec{E}$ since a has no loop. $j \in \{2, ..., |C^1| - 2\}$ hence one can use either (a, c_j^1) or (c_j^1, a) to build a cycle shorter than C^1 containing a, u and v, which would therefore be a cycle that would have all the required properties. This is impossible by minimality of C^1.

- There can be cross-bridges between C^1 and C^2. However, a subtle line of reasoning explained below shows that here, keeping the indices used previously, we only have to avoid the cross-bridges $[(c^1_i, v)$ and $(v, c^1_{i+1}) \in \vec{E}$ with $c^1_i \neq v \neq c^1_{i+1}]$ with $i = i_0 - 1$ and $i = i_0 + 2$ in order not to disturb how the generic construction from Section 6.1.3 works.

  The case $i = i_0 - 1$ is impossible because $(v, c^1_{i_0}) = (v, u) \notin \vec{E}$. If $(v, c^1_{i_0+3}) \in \vec{E}$ then we can use it to find a cycle shorter than C^1 that contains everything we want – unless $c^1_{i_0+2} = a$.

  If $c^1_{i_0+2} = a$ then we redefine $C^2 = \{u\}$ with which all that we have proved can be adapted: we only have to avoid the cross-bridges $[(c^1_i, v)$ and $(v, c^1_{i+1}) \in \vec{E}$ with $c^1_i \neq v \neq c^1_{i+1}]$ with $i = i_0 - 2$ and $i = i_0 + 1$. Once again, this is impossible except if we also have $c^1_{i_0-1} = a$. Then C^1 is made of only three elements and this case is solved in Section 6.1.8.

- As we will see, there is only one possibility for C^1 to have both an attractive and a repulsive vertex. Since $C^2 \subset C^1$, it is enough to consider attractive and repulsive vertices for C^1 that are located in C^1. Let t be an attractor located in C^1 for all elements of C^1. Notably, $(a, t) \in \vec{E}$. Since a has no loop, this Case 1.3 causes $(t, a) \in \vec{E}$. Similarly, for p a repulsive vertex, we have not only $(p, a) \in \vec{E}$, but also $(a, p) \in \vec{E}$. Hence the shortest cycle that meets all our requirements is (a, p, u, v, t), so the only possibility for C^1 to have both that does not contradict its minimality is to be this precise cycle (with some of the vertices being possibly equal). This is a case we treat in Section 6.1.8.

Cross-bridges remark for Case 1.3: We prove why we only need to avoid two cross-bridges. The basics of Case 1.3 are: $(u, v) \in \vec{E}$, $(v, u) \notin \vec{E}$, a loopless, C^1 is a possibly self-intersecting cycle made of the concatenation of a path from a to u and one from v to a, all unidirectional edges are from or to a vertex with a loop, $C^2 = \{v\}$.

Any diagonal region that contains the coding of a tile, delimited by buffers and possibly border slices above or below, does encode exactly one tile, see Fig. 6.5. Indeed, each of its vertical slices contains at least one of the elements among $\{c^1_{i_0}, c^1_{i_0+2}\}$ (since the buffer is given by $c^1_{i_0+1}$), and these are always part of a micro-slice that encodes something. By construction, the whole slice encodes the same tile number, vertically.

Moreover, two adjacent slices also contain the same encoding, using the fact that there is no cross-bridge for $i = i_0 - 1$ or $i = i_0 + 2$. Indeed, suppose we have a $c^1_{i_0}$ coding micro-slice in the rightmost slice. To its left, in the second rightmost slice, is a $c^1_{i_0-1}$ coding micro-slice that encodes the same thing, since there is no cross-bridge for $i = i_0 - 1$. Since we force two vertically adjacent coding micro-slices to encode the same tile if none of them is a buffer, the coding micro-slice using $c^1_{i_0}$, that is below the one using $c^1_{i_0-1}$, encodes the same tile as the latter. But to the left of the $c^1_{i_0}$ coding micro-tile is a $c^1_{i_0-1}$ coding micro-tile that encodes the same tile. Below this one is, once again, a $c^1_{i_0}$ coding micro-tile that encodes the same tile by construction... The same reasoning works when starting from a $c^1_{i_0+2}$ coding micro-slice in the leftmost slice: it encodes the same thing as the $c^1_{i_0+3}$ coding micro-slice to its right since there is no cross-bridge for $i = i_0 + 2$, etc.

Overall, in this case 1.3 one of the following happens:

- the C^1 and C^2 we chose fulfill condition C;
- or C^1 has both an attractive and a repulsive vertex, and so is of length 5 – this is treated in Section 6.1.8;

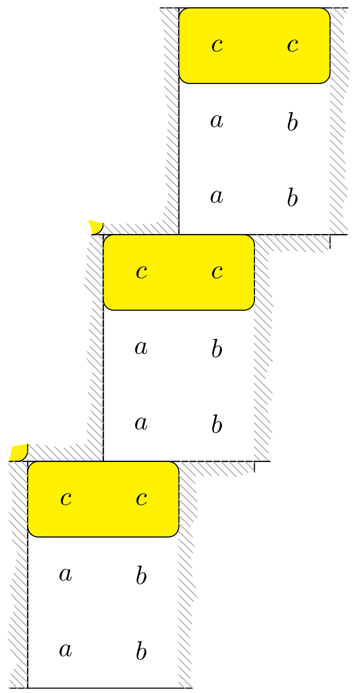


Figure 6.5: A coding region reused from Fig. 6.4, delimited by buffers, with an example of transmission that is ensured between the different slices of it.

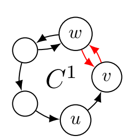
- or the only condition missing is that there are cross-bridges. Here, if C^1 has 4 or more elements, we prove that we can reproduce what matters of the generic construction in spite of this in this specific case, notably due to the line of reasoning above and because one specific cross-bridge is still impossible;
- or C^1 is of length 3, and this is solved in Section 6.1.8.

Consequently, we solved case 1.3.

6.1.6 Case 2

Here, we assume that $\widetilde{\mathcal{G}}(H)$ contains no loop but at least one bidirectional edge.

Case 2.1: we can find one simple cycle of length at least 3, with at least one bidirectional edge.



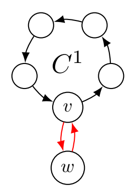
Focus on a cycle that contains both a bidirectional and a unidirectional edge – we can find one, else $\widetilde{\mathcal{G}}(H)$ would be of symmetric type. Just as in Case 1.2 from Section 6.1.5, reduce it iteratively so that it ends up either in a graph – as small as possible – similar to the one of Case 2.2 and is therefore treated similarly; or as the smallest cycle with both a bidirectional and a unidirectional edge forming a graph that does not contain a subgraph from Case 2.2.

Name C^1 this cycle; name u, v and w some successive vertices in C^1 such that $(u, v), (v, w)$ and $(w, v) \in \vec{E}$ but $(v, u) \notin \vec{E}$; and define C^2 as the cycle made of only v and w. Also, define $c^1_0 = v = c^2_0$.

As in Case 1.2, notice that all edges from w must lead either to c^1_2 or to $v = c^1_0$, else we could find a shorter cycle C^1. There remains to check that C^1 and C^2 have the properties we want:

- $|C^1| \geq 3$;
- C^1 and C^2 have a good pair, that is (c^1_2, v);
- C^2 has no uniform shortcut since it is of length 2 with no loop. If C^1 had a uniform shortcut, it could not be of size 0 (because all vertices are loopless) or of size -1 (because $(v, u) \notin \vec{E}$). Any other size of shortcut is impossible due to the aforementioned property of w.
- If there was a cross-bridge, we reach a contradiction in the exact same fashion as what is done in the case of a cross-bridge in Case 1.2.
- There cannot be any attractive or repulsive vertex for C^1 located in C^1 since no vertex has a loop in the present case. None can be located in C^2 either since $C^2 \subset C^1$.

Case 2.2: any simple cycle of length at least 3 contains no bidirectional edge.



Since there are bidirectional edges in $\widetilde{\mathcal{G}}(H)$ (hypothesis of Case 2), which is strongly connected, we can find at least one (simple) cycle of length ≥ 3 that has one vertex in common with a bidirectional edge. Choose a minimal cycle among these ones, call it C^1; name v the vertex it has in common with the bidirectional edge, and w the other vertex. We define C^2 as the cycle containing only v and w. Call $c^1_0 = v = c^2_0$.

- $|C^1| \geq 3$;
- (c^1_1, w) is a good pair for C^1 and C^2;
- C^2 has no uniform shortcut since it is of length 2 with no loop. If C^1 had uniform shortcuts, they could not be of length 0 because none of its vertices has a loop; they could not be of length -1 because none of its edges is bidirectional; and they could not be of any other length else the shortcut starting from v would allow us to define a strictly shorter cycle with the same property, contradicting the minimality of C^1.
- If there was a cross-bridge, we could have two cases:

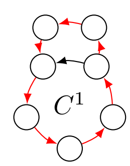
- First is (c_i^1, v) and $(w, c_{i+1}^1) \in \vec{E}$, with $c_i^1 \neq w$ and $v \neq c_{i+1}^1$. Then we could use the edge (w, c_{i+1}^1) for the following cycle: $\{w, c_{i+1}^1, c_{i+2}^1, ..., v\}$. It would be of length at least 3 since $c_{i+1}^1 \neq v$, no vertex would repeat, and it would contain one bidirectional edge. This is impossible by assumption of Case 2.2.
- Second is (c_i^1, w) and $(v, c_{i+1}^1) \in \vec{E}$, with $c_i^1 \neq v$ and $w \neq c_{i+1}^1$. Then we could use the edge (v, c_{i+1}^1) to define a cycle strictly shorter than C^1, with the same properties, of length at least 3 (c_{i+1}^1 cannot precede v, else we would have a bidirectional edge). This is impossible by minimality of C^1.

- There is no attractive or repulsive vertex for C^1 located in C^1 since none has a loop. But it seems that there can be an attractive and/or repulsive vertex for C^1 located in C^2, that is, w. Nevertheless, if w was both attractive and repulsive, using part of C^1 we could build a (simple) cycle of length at least 3 including v and w, hence a bidirectional edge. This is forbidden in this Case 2.2.

6.1.7 Case 3

In this subsection, we assume $\widetilde{\mathcal{G}}(H)$ contains no loop and no bidirectional edge.

Case 3.1: considering the smallest cycle C in $\widetilde{\mathcal{G}}(H)$ one can find, there exists a path γ between two different vertices of C that does not intercept C elsewhere, and γ is of a length different from the length between these vertices inside C.



Define C^1 a cycle with said property for a path γ between two of its vertices so that C^1 is a cycle of minimal length. If there are several cycles of minimal length, choose one so that we can find a path γ as short as possible. Now, we name the vertices: γ is a path between c_0^1 and c_k^1, that is not of length k. Define C^2 as the concatenation of γ and $(c_k^1, c_{k+1}^1, ..., c_0^1)$, with $c_0^2 := c_0^1$ and $c_l^2 = c_k^1$ with $l = |\gamma| > k$.

- $|C^1| \geq 3$ since there is no bidirectional edge;
- (c_1^1, c_1^2) is a good pair;
- In both C^1 and C^2, we have no uniform shortcut of length 0 or -1 since there are no loops and no bidirectional edges. C^1 cannot have any other length of uniform shortcut, or even any edge between two of its vertices, else we could find a strictly smaller cycle with a path of a different length between two of its vertices.

  Suppose C^2 has a uniform shortcut of length j. The point if it happens is to build a new cycle, C^3, so that C^1 and C^3 work for the generic construction in Section 6.1.3. If we have an uniform shortcut, then $c_0^2 = c_0^1$ cannot lead to an element that C^2 shares with C^1 by minimality of the latter, hence the edge of its shortcut must lead to some c_j^2, with $1 < j < l$ the uniform length of the shortcuts. Necessarily, $(c_0^1 = c_0^2, c_j^2, c_{j+1}^2, ..., c_l^2 = c_k^1)$ being a path between two elements of C^1 that is strictly shorter than γ, we have $l-j+1=k$ in order not to reach a contradiction. Hence $j=l-k+1$. Then $C' := (c^2_0, c^2_{l-k+1}, c^2_{l-k+2}, ..., c^2_l = c^1_k, c^1_{k+1}, ..., c^1_{-1})$ is a cycle of length $|C^1|$.

First, we study the case $k \neq 1$. Then $(c^2_0, c^2_1, c^2_2, ..., c^2_{l-k+1})$ is a path of length $l-k+1 < l$, linking two elements of C' (these elements are c^2_0 and c^2_{l-k+1}, which are consecutive in C'). Since C' is of length $|C^1|$ and we found a path of length smaller than l joining two of its vertices, this fact contradicts the minimality of C^1, so we cannot actually have $k \neq 1$.

Necessarily $k=1$. Then $j=l$, and the edge between c^2_0 and $c^2_l = c^1_k = c^1_1$ is already part of C^1 – it is (c^1_0, c^1_1). We have $l > k$ so $l \neq 1$; if $l=2$ then $j=2$ so $c^2_2 = c^1_1$ would have an edge going to $c^2_4 = c^1_3$, an element of C^1 necessarily (possibly c^1_0). This is impossible by minimality of C^1. We deduce that $l > 2$.

We set $C^3 := (c^2_0, c^2_1, c^2_2, c^1_3, c^1_4, ..., c^1_{|C^1|-1})$ using the edge (c^2_2, c^2_{l+2}) since $j=l$, which is the edge (c^2_2, c^1_3) since $c^2_l = c^1_1$. There is a specific case if $c^1_3 = c^1_0 = c^2_0$, where both C^1 and C^3 end up being triangles, but the reasoning below still holds.

Instead of using C^1 and C^2, we check that choosing C^1 and C^3 – with the latter in lieu of C^2 in Section 6.1.3 – for our generic construction works well:

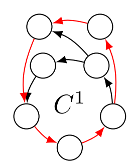
- $|C^1| \geq 3$, this does not change;
- (c^1_1, c^2_1) is still a good pair for C^1 and C^3;
- C^1 being still defined the same way, it does not contain any uniform shortcut or even any edge between two of its vertices. If C^3 contains uniform shortcuts, the one starting at $c^1_0 = c^2_0$ must lead to c^2_2 since it must be of size different from 1 and it must not lead to an element in common with C^1, because the latter is minimal. But if $(c^2_0, c^2_2) \in \vec{E}$, then we could find a path strictly shorter than γ between c^1_0 and c^1_1, that would not be of length 1 (because $c^2_2 \neq c^2_l$). This contradicts the minimality of C^1.
- Since no edge between two non-consecutive elements of C^1 is possible, the unique cross-bridge between C^1 and C^3 would be some (c^1_i, c^2_2) and $(c^2_1, c^1_{i+1}) \in \vec{E}$. But it would also be a cross-bridge between C^1 and C^2, and this is in all cases impossible, see below.
- There is no attractive or repulsive vertex for C^1 in C^1 since no element of C^1 has a loop. Suppose there are both an attractive and a repulsive vertex for C^1 located in $C^3 \setminus C^1$, call them t^3 and p^3. They must be distinct (because the graph has no bidirectional edge) and not be in C^1; hence C^3 contains at least 2 exclusive vertices.

  Then the idea is to use C^1 as "the C^2" of the generic construction, and C^3 as "the C^1" of the generic construction: $|C^3| \geq 3$, and all of our other properties hold when using C^3 as "the C^1" in checking condition C, except possibly the attractive and repulsive conditions. Hence the only facts that we have to verify to exchange their roles is that there is no attractive or repulsive vertex for C^3 located either in C^3 or in C^1. There is none in C^3 because no vertex of C^3 has a loop. If there was both an attractive and a repulsive vertices for C^3 in C^1, call them t^1 and p^1, then notably t^1 would lead to t^3 and vice versa... But no bidirectional edge exists here. So, up to exchanging what is C^1 and what is C^3, we cannot have both an attractive and a repulsive vertex for C^1. This reasoning will be applied again and be called the *trick of exchanging the roles.*

Therefore in the worst case, if there are uniform shortcuts in C^2, we can build the generic construction from Section 6.1.3 with the present C^1 (as C^1 in the generic construction) and C^3 (as C^2 in the generic construction).

- Suppose we have a cross-bridge, that is, (c_i^1, c_{j+1}^2) and $(c_j^2, c_{i+1}^1) \in \vec{E}$. Since C^1 is minimal, c_j^2 and c_{j+1}^2 are not elements of C^1, so $j+1 < l$. Then $(c_0^1 = c_0^2, c_1^2, ..., c_j^2, c_{i+1}^1)$ is a path between two elements of C^1 that is necessarily strictly shorter than γ since $j+1 < l$ (with $c_l^2 = c_k^1$). This is possible only if the obtained path is of length $i+1$, the same length as the one between c_0^1 and c_{i+1}^1 in C^1. It would mean that $j = i$, but then $(c_i^1, c_{j+1}^2, c_{j+2}^2, ..., c_l^2 = c_k^1)$ would also be a path γ' shorter than γ, and the only possibility is then that $k - i + 1 = l - j + 1$ (the distance between c_i^1 and c_k^1 is the length of γ'), from which we deduce $k = l$, which is impossible.
- Once again, for attractive and repulsive vertices we use the trick of exchanging the roles of C^1 and C^2 – this holds here notably because $|C^2| \geq 3$.

Case 3.2: considering the smallest simple cycle C in $\widetilde{\mathcal{G}}(H)$, any path γ we can find between two different vertices of C that does not intercept C elsewhere is of the same length as the length between these vertices inside C; and we can find at least one such path γ.



Define $C^1 := C$. We use the following algorithm: we start with $V_0 := \{c_0^1\}$ and $(V_i)_{i \in [1, |C^1|]}$ empty. Then we recursively append to V_{i+1} all vertices w in $\widetilde{\mathcal{G}}(H)$ so that there is a $v \in V_i$ with $(v, w) \in \vec{E}$, where the index is considered modulo $|C^1|$ so that $V_{|C^1|} = V_0$. The algorithm halts when it tries to append vertices to a V_i that are all already in it, which happens because $\widetilde{\mathcal{G}}(H)$ is made of a finite number of vertices. The fact that no path exterior to C^1 is of a different length than the corresponding path C^1, plus the absence of any loop or bidirectional edge, makes all the V_i disjoint. Finally, the strong connectivity we assumed ensures that $H = \bigsqcup_{i=0}^{|C^1|-1} V_i$.

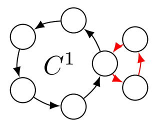
We use the fact that H does not verify condition D, specifically is not of state-split cycle type. Since by construction, for any $v \in V_i$, we have $(v, w) \in \vec{E} \Rightarrow w \in V_{i+1}$, the only possibility is that there exist $v \in V_{i_0}$ and $w' \in V_{i_0+1}$ so that $(v, w') \notin \vec{E}$. However, we also have some $v' \in V_{i_0+1}, (v, v') \in \vec{E}$ and $w \in V_{i_0}, (w, w') \in \vec{E}$. Obviously, the four vertices are different. Now, take a (simple) path γ_1 from v' to v that is as short as possible. Take a different (simple) path γ_2 from w' to w that still has as much of its trajectory in common with γ_1 as possible. We redefine C^1 as the concatenation of γ_1 and (v, v') and C^2 as the concatenation of γ_2 and (w, w').

It is rather easy to see that the properties we need for our generic construction are verified:

- $|C^1| \geq 3$, since there is no bidirectional edge;
- The unique common part between C^1 and C^2 is the biggest sequence of vertices γ_1 and γ_2 have in common, so starting from the first pair on which they disagree we obtain a good pair;
- There is no uniform shortcut between C^1 and C^2 if γ_1 was chosen minimal and γ_2 as close to γ_1 as possible;

- There is no cross-bridge for the same reason;
- For attractive and repulsive vertices, we use the trick of exchanging the roles of C^1 and C^2 described in Case 3.1.

Case 3.3: considering the smallest simple cycle C in $\widetilde{\mathcal{G}}(H)$, we can find no path between two different vertices of this cycle.



Since $\widetilde{\mathcal{G}}(H)$ does not verify condition D, it cannot be a simple cycle, hence a path exists from one vertex of C to itself that does not intersect C elsewhere. Define $C^1 := C$ and this vertex as c_0^1. Then, considering the smallest path γ from c_0^1 to itself outside of C^1, we define $C^2 := \gamma$, with $c_0^2 := c_0^1$.

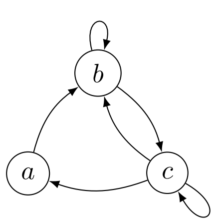
It remains to check that these C^1 and C^2 verify the properties we need.

- $|C^1| \geq 3$;
- C^1 and C^2 have exactly one vertex in common, $c_0^1 = c_0^2$, and so (c_1^1, c_1^2) is a good pair;
- There is no uniform shortcut of length 0 or -1 neither in C^1 nor in C^2, since there are no loop and no bidirectional edge. There is no uniform shortcut of any other length; else consider the edge starting at c_0^1, be it in C^1 or in C^2: it would allow us to build a shorter C^1 or a shorter C^2, contradicting the fact that the two of them have been chosen to be minimal.
- There is no cross-bridge between C^1 and C^2 because it would allow us to build path outside of C^1 between two distinct elements of C^1, which is impossible in this case;
- There is no attractive or repulsive vertex for C^1 in C^1, because no element of C^1 has a loop. There is no attractive or repulsive vertex for C^1 in C^2 else we could build a path outside of C^1 between two distinct elements of C^1, which is impossible in this case.

6.1.8 Additional cases

Most three-vertex cases

For most three-vertex graphs, we can apply the generic construction from Section 6.1.3 without any problem. However, some of them require to be slightly more cautious, because some properties from condition C (Definition 6.3) are missing. Among the four of them up to a change of labels, the first three are the following:

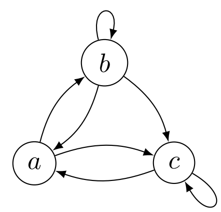


Here, there are both an attractive and a repulsive vertex. Still, similarly to case 1.1 from Section 6.1.5, a has in-degree 1 since only c leads to a, and so Proposition 6.5 holds, since a forces the alignment of columns.

Besides, we have to be careful about the cross-bridge property. In the first example, we choose $C^2 = (b)$ (there is no cross-bridge then because $(b, a) \notin \vec{E}$). In the second and in the third, we choose $C^2 = (a, c)$ (a has no loop hence there is no problem of cross-bridge or of uniform shortcut in C^2). All the other properties from Condition C are verified.

One specific three-vertex case

Among three-vertex graphs remains the following one:



Here we have three problems:

- We could have cross-bridges, so to avoid them we choose $C^2 = (a, b)$ to perform the proofs that follow (we have no problem of uniform shortcut in C^2 since a has no loop);
- We have attractive and repulsive vertices;
- And here, contrary to Case 1.1, we cannot rely on an element of the alphabet that must be followed or preceded by a specific other one to solve that problem.

The reasoning is slightly subtler than in the previous subsubsection then. Try to perform the proof of propositions Proposition 6.5 and Proposition 6.6 by taking two successive columns K_1 and K_2. In any macro-slice of K_1, there is some a meso-slice (made of NM symbols a) that is vertically preceded by a c meso-slice. The a meso-slice must not be next to any symbol a in column K_2. But then if there is any c in the part of K_2 horizontally adjacent to this a meso-slice, the aforementioned c meso-slice of K_1 is horizontally followed by at least one b (be it from a regular meso-slice, a border, or a code); but this cannot be. Hence an a meso-slice in K_1 can only be horizontally followed by symbols b. So two columns are always aligned even if we have attractive and repulsive vertices.

The rest of the generic construction works normally.

Specificity of Case 1.3

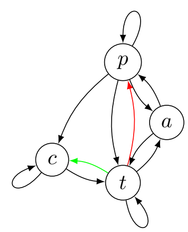
We focus on specific subcases where $C^2 = (c)$ and $C^1 = (a, p, b, c, t)$ where all vertices must not necessarily be different, with t attractive, p repulsive, $(c, b) \notin \vec{E}$, loops on b and c, and a loopless. Additionally, the initial and terminal vertices of any unidirectional edge must have a loop. Five cases can happen; here we treat only the fourth one, and all the others except the last one are done similarly: said fifth case is treated among length-3 cases.

- All elements are distinct;

- $p = t$ and all others distinct;
- $c = t$ and all others distinct;
- $b = p$ and all others distinct;
- $b = p$ and $c = t$: this is one of the three-vertex graphs we have seen before.

Note that in all those cases, stemming from the analysis performed in case 1.3 in Section 6.1.5, most of condition C (Definition 6.3) seems to hold except for the presence of both attractive and repulsive vertices, that endangers Proposition 6.5. The only fact that we have to check is that we can circumvent this obstacle in a way similar to what is done with the length-3 cases.

If $b = p$, we obtain the following graph:



We added the red edge so that we cannot reduce the graph to a strictly smaller cycle (with three vertices) on which we already proved the generic construction worked. If there was an edge between a and c, it would be bidirectional (since a has no loop). But since the edges between c and t or c and p cannot be both bidirectional, we could reduce the present cycle to a strictly smaller one containing a unidirectional edge, a loop and a loopless vertex. So there is no edge between a and c. Since $p = b$ there is none from p to c. The only optional edge available is (t, c) (in green).

Try to perform the proof of Proposition 6.5 and take two horizontally successive columns K_1 and K_2. In any macro-slice of K_1, there is a c meso-slice slice that is above a t meso-slice, itself above a a meso-slice.

The a meso-slice in K_1 cannot be horizontally followed by a border or a code meso-slice in K_2 because they contain c. The c meso-slice in K_1 must not be followed horizontally by any symbol p or a in column K_2 – so most of it (at least $NM/2$) is in contact neither with a border nor with a code meso-slice, but with a meso-slice made of only one symbol. If that symbol is c, then the aforementioned a meso-slice is in contact with a meso-slice made of a symbol a. This is impossible. Hence the c meso-slice of K_1 is mostly followed by a t meso-slice of K_2, and from this we recover Proposition 6.5.

The three other cases are treated similarly, exploring with what each C^1 meso-slice can be in contact to ensure that Proposition 6.5 is valid even without all of condition C. Checking the rest of the properties follows case 1.3.

6.1.9 Proof for several strongly connected components

Suppose that the Rauzy graph of H of order 1 is made of several SCCs and does not verify condition D. Either one of them belongs to none of the types listed in Definition 6.1, and we can simply reduce H to that SCC and perform the previous construction with it; or each of them belongs to at least one of the three types listed in Definition 6.1. In that second case, we study $H \times H \times H$ instead, and prove that this one's Rauzy graph of order 1 has a SCC *made of triplets of distinct elements of* $\mathcal{A}_H$ that is none of the three types that constitute condition D. We can then apply Theorem 12 to $H \times H \times H$, and for any two-dimensional SFT Y we can find $V' \subseteq (\mathcal{A}_H \times \mathcal{A}_H \times \mathcal{A}_H)^{\mathbb{Z}}$ so that $X_{H\times H\times H,V'}$ is a root of Y. Now, "spread" each cell of $X_{H\times H\times H,V'}$ into three adjacent vertical cells containing the three coordinates of the product alphabet, and consider the subshift X on ${\mathcal{A}_H}^{\mathbb{Z}}$ obtained by taking the topological and orbit closure from this. It is by construction a $(1,3)$-root of $X_{H\times H\times H,V'}$ – note that it is important for the three coordinates to be distinguishable, and it is the case as they have no symbol in common. Consequently, X is a root of Y too. Furthermore, X is some X_{H,V_Y} where V_Y is a SFT that can be computed from V' by "spreading" the three coordinates of the cells of the latter into three vertically adjacent cells of the former. Constraints on V' are then passed onto V naturally, notably because the symbols used for the different coordinates in V' are disjoint.

Thus we computed algorithmically some V_Y so that the SFT $X = X_{H,V_Y}$ is a root of Y.

Now, the only point left in this reasoning is to show the existence of the desired SCC in $H\times H\times H$.

Since H does not verify condition D, it has a non-reflexive SCC S_1, a non-symmetric SCC S_2 and a non-state-split SCC S_3. Suppose the three of them are distinct. Then:

- Since S_1 is non-reflexive, no SCC of $S_1 \times S_2 \times S_3$ is reflexive. Indeed, since S_1 is strongly connected, all vertices of S_1 are represented in any SCC C of that graph product, meaning that for any $s_1 \in S_1$ there is at least one vertex of the form $(s_1, *, *)$ in C. But if C had loop on all its vertices, then in particular S_1 would be reflexive.

- Similarly, since S_2 is non-symmetric, no SCC of $S_1 \times S_2 \times S_3$ is symmetric.

- Finally, since S_3 is non-state-split, no SCC of $S_1 \times S_2 \times S_3$ is a state-split cycle. Indeed, suppose S is such a state-split SCC of the direct product. It can be written as a collection of classes $(V_i)_{i\in I}$ of elements from $S_1 \times S_2 \times S_3$ that we can project onto S_3, getting new classes $(W_i)_{i\in I}$, with some elements of S_3 that possibly appear in several of these. Let c be any vertex in S_3 that appears at least twice *with the least difference of indices between two classes where it appears*; say $c \in W_i$ and $c \in W_{i+k}$. Since S is state-split, all elements in W_{i+1} are exactly the elements of S_3 to which c leads. But it is the same for W_{i+k+1}. Hence $W_{i+1} = W_{i+k+1}$. From this we deduce that $W_i = W_{i+k}$ for any i, using the fact that indices are modulo $|I|$. Since k is the smallest possible distance between classes having a common element, classes from $(W_i)_{i\in\{0,\dots,k-1\}}$ are all disjoint; and they obviously contain all vertices from S_3. Now simply consider these classes W_0 to W_{k-1}: you get the proof that S_3 is state-split.

If two of the SCCs S_1, S_2 and S_3 are the same, perform the whole reasoning with $H \times H$ instead of $H \times H \times H$.

6.2 Properties of two-dimensional subshifts under interplay

6.2.1 Periodicity

For $x \in X$ with X a two-dimensional subshift, we say that x is *periodic* (of period $\vec{v}$) if there exists $\vec{v} \in \mathbb{Z}^2 \setminus \{(0,0)\}$ such that $\forall (i,j) \in \mathbb{Z}^2, x_{(i,j)} = x_{(i,j)+\vec{v}}$. In a more general setting of SFTs on groups, this is called weak periodicity – see Definition 7.6.

Corollary 6.8. *Let H be a one-dimensional nearest-neighbor SFT.*

$$X_{H,V} \text{ is empty or contains a periodic configuration for all one-dimensional SFTs } V \Leftrightarrow \widetilde{\mathcal{G}}(H) \text{ verifies condition } D.$$

Proof. If $\widetilde{\mathcal{G}}(H)$ verifies condition D, then, as is detailed in the proof of Theorem 13, whatever may be the chosen V, we can find a patch P that respects the local rules of $X_{H,V}$ and tiles the plane periodically. Hence $X_{H,V}$ admits a periodic configuration.

If $\widetilde{\mathcal{G}}(H)$ does not verify condition D, then, using Theorem 12, we know that for any two-dimensional SFT Y with no periodic configuration, there exists some one-dimensional SFT V_Y such that X_{H,V_Y} is a (m,n)th root of Y.

We consider Y a two-dimensional SFT with no periodic configuration (see [Rob71] for instance). Then, naming V_Y the corresponding one-dimensional SFT from Theorem 12, we know that there exists $\psi\colon Y \hookrightarrow X_{H,V}$ continuous with $\sqcup_{0\leq i<m, 0\leq j<n}\sigma^{(i,j)}(\psi(Y)) = X_{H,V}$ for some integers m and n. Note that we use ψ the inverse map of φ in the definition of a (m,n)th root, the reasoning being easier with it.

If $X_{H,V}$ contained a σ-periodic configuration, then $\sqcup_{0\leq i<m, 0\leq j<n}\sigma^{(i,j)}(\psi(Y))$ would, and so $\psi(Y)$ would too (since configurations in $\sqcup_{0\leq i<m, 0\leq j<n}\sigma^{(i,j)}(\psi(Y))$ are merely translates of the ones in $\psi(Y)$).

Call $\psi(y)$ such a periodic configuration, with $y \in Y$. There exists some $\vec{v} = (a,b) \in \mathbb{Z}^2$ such that $\sigma^{\vec{v}}(\psi(y)) = \psi(y)$. But consequently, $\sigma^{mn\vec{v}}(\psi(y)) = \psi(y)$. Then $\sigma^{(anm,bmn)}(\psi(y)) = \psi(y)$. Using Theorem 12 again, we obtain that $\psi(\sigma^{(an,bm)}(y)) = \psi(y)$. ψ being bijective, we finally get:

$$\sigma^{(an,bm)}(y) = y.$$

We found a periodic configuration in Y. This being impossible, we conclude that $X_{H,V}$ contains no periodic configuration. □

6.2.2 The Domino Problem with interplay

From Section 6.1, we deduce:

Theorem 13. *Let H be a nearest-neighbor one-dimensional SFT.*

$$DP_I(H) \text{ is decidable} \Leftrightarrow \widetilde{\mathcal{G}}(H) \text{ verifies condition } D.$$

Proof. The following proof is similar to the one of Theorem 9.

Proof of $\Leftarrow$: assume $\widetilde{\mathcal{G}}(H)$ verifies condition D. Then its SCCs share a common type, be it reflexive, symmetric, or state-split cycle. For each of these three cases, we produce an algorithm that takes as input a one-dimensional SFT $V \subset \mathcal{A}^{\mathbb{Z}}$, and that returns `YES` if $X_{H,V}$ is nonempty, and `NO` otherwise.

Let M be the maximal size of forbidden patterns in $\mathcal{F}_V$ (since V is an SFT, such an integer exists).

- If $\widetilde{\mathcal{G}}(H)$ has state-split cycle type SCCs: let L be the LCM of the number of V_is (whose disjoint union forms a component) in each of the strongly connected components. If there is no rectangle of size $L \times M(|\mathcal{A}|^{LM}+1)$ respecting local rules of $X_{H,V}$ and containing no letter that corresponds to a transient vertex in $\widetilde{\mathcal{G}}(H)$, then answer `NO`. Indeed, any configuration in

$X_{H,V}$ contains valid rectangles as large as we want that do not contain such transient letters. If there is such a rectangle R, then by the pigeonhole principle it contains at least twice the same rectangle R' of size $L \times M$. To simplify the writing, we assume that the rectangle that repeats is the one of coordinates $[1, L] \times [1, M]$ inside R where $[1, L]$ and $[1, M]$ are intervals of integers, and that it can be found again with coordinates $[1, L] \times [k, k + M - 1]$. Else, we simply truncate a part of R so that it becomes true.

Define $P := R|_{[1,L]\times[1,k+M-1]}$. Since V has forbidden patterns of size at most M, and since R respects our local rules, P can be vertically juxtaposed with itself (overlapping on R').

P can also be horizontally juxtaposed with itself (without overlap). Indeed, one line of P uses only elements of one SCC of H (since elements of two different SCCs cannot be juxtaposed horizontally, and we banned transient letters). Since L is a multiple of the length of all cycle classes, the first element in a given line can follow the last element in the same line. Hence all lines of P can be juxtaposed with themselves.

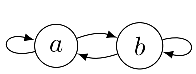
As a conclusion, P is a valid patch that can tile $\mathbb{Z}^2$ periodically. Therefore, $X_{H,V}$ is nonempty; return `YES`.

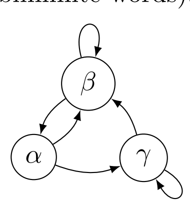
- If $\widetilde{\mathcal{G}}(H)$ has symmetric type SCCs the construction is similar, but this time build a rectangle R of size $2 \times M(|\mathcal{A}|^{2M} + 1)$. Either we cannot find one and return `NO`; or we can find one and from it extract a patch that tiles the plane periodically and return `YES`.

- Finally, if $\widetilde{\mathcal{G}}(H)$ has reflexive type SCCs, the construction is even simpler than before. Build a rectangle R of size $1 \times M(|\mathcal{A}|^{M} + 1)$; the rest of the reasoning is identical.

Proof of $\Rightarrow$ is due to Theorem 12, and is done by contraposition. If $\widetilde{\mathcal{G}}(H)$ does not verify condition D, then for any Wang shift W we can algorithmically build some one-dimensional SFT V_W such that X_{H,V_W} is a root of W, see Theorem 12. If we were able to solve $DP_I(H)$, then there would exist a Turing Machine $\mathcal{M}$ able to tell us if $X_{H,V}$ is empty for any one-dimensional SFT V. But as a consequence we could build a Turing Machine $\mathcal{N}$ taking as input any Wang shift W, and building the corresponding V_W following Section 6.1. Then, by running $\mathcal{M}$, $\mathcal{N}$ would be able to tell us if X_{H,V_W} is empty or not. Since X_{H,V_W} is empty if and only if W is, it could consequently answer if W is empty or not; but determining the emptiness or nonemptiness of every Wang shift is equivalent to $DP(\mathbb{Z}^2)$ being decidable, which is false. Hence, since $DP(\mathbb{Z}^2)$ is undecidable, $DP_I(H)$ is too. □

Remark 32. A pair of conjugate SFTs H_1 and H_2 may yield different results, with $DP_I(H_1)$ decidable but $DP_I(H_2)$ undecidable. Consider for instance the following Rauzy graphs and applications on finite words (extensible to biinfinite words):





$$\varphi : \begin{cases} aa \mapsto \gamma \\ ab \mapsto \beta \\ ba \mapsto \alpha \\ bb \mapsto \beta \end{cases} \qquad \psi : \begin{cases} \alpha \mapsto a \\ \beta \mapsto b \\ \gamma \mapsto a \end{cases}$$

These graphs describe conjugate SFTs through these applications. However, the first graph has decidable $DP_I(H)$ and the second has not, by use of Theorem 13.

6.3 Impact of the interplay on the entropy

6.3.1 Horizontal constraints without condition D

From the construction in Section 6.1 we deduce the following:

Proposition 6.9. *Let H be a one-dimensional nearest-neighbor SFT that does not satisfy condition D. Then there exists a one-dimensional SFT V such that $h(X_{H,V})$ is not computable.*

Proof. Let W be a Wang shift with a non-computable entropy. By using Theorem 12 and Proposition 5.4, there exists a one-dimensional SFT V such that

$$h(W) = KM^2Nh(X_{H,V})$$

where N is the number of elements in the alphabet of W, and K and M are defined as in Section 6.1.3 and depend only of H – all of them being integers, and consequently computable. Therefore, $h(X_{H,V})$ is not computable. □

Remark 33. For a given H, the V-dependent entropies in general are still not characterized and it seems difficult since in the construction of the root, the entropy decreases if the number of Wang tiles increases. In fact the previous proof allows to expect that in the case where the condition D is not satisfied, there exists a constant C_H such that for every Π_1^0-computable number h smaller than C_H there exists a vertical one-dimensional subshift V that allows $h(X_{H,V}) = h$. However, we do not obtain this result exactly since if the Kolmogorov complexity of h is important, then the cardinality of the alphabet of the Wang subshift that has h as entropy, named N in the previous proof, is also important.

Nevertheless, in the case where there are cycles in the Rauzy graph which defines H that do not appear in the coding of the Wang subshift used in Theorem 12, we can encode h in a given part of $X_{H,V}$, diluted by the necessarily large macro-slices, but then add a noisy zone where these cycles can be used to increase the entropy, as it is done in the proof of Theorem 11.

6.3.2 Condition D, computable entropy

A one-dimensional SFT H that verifies condition D can yield computable entropies. Indeed, one has the immediate result that allows only a small range of available entropies:

Proposition 6.10. *The entropies $h(X_{\mathcal{A}^{\mathbb{Z}},V})$ accessible for $V \subset \mathcal{A}^{\mathbb{Z}}$ SFT are all the entropies accessible for one-dimensional SFTs with alphabet $\mathcal{A}$. These are included in the values $\log_2(\lambda) \leq \log_2(|\mathcal{A}|)$, where λ is a Perron number. Notably, they all are computable.*

Proof. We use the fact that $N_{X_{\mathcal{A}^{\mathbb{Z}},V}}(n,n) = N_V(n)^n$ since any two columns of height n can be juxtaposed horizontally here. We conclude using [LM95] that states that the available entropies for one-dimensional SFTs are the $\log_2(\lambda)$ where λ is a Perron number. □

Remark 34. An open question remains: what exactly are these accessible $\log_2(\lambda)$ obtained for a fixed size of alphabet? Corollary 5.8 – and [HM10] – gives an answer for dimension 2, but this exact question is, to our knowledge, not answered in dimension 1.

A similar result holds for a larger class of graphs, one of the possibilities for respecting condition D:

Proposition 6.11. *Let $H \subset \mathcal{A}^{\mathbb{Z}}$ be a nearest-neighbor SFT whose Rauzy graph is so that each of its SCCs is a state-split cycle. Then for all $V \subset \mathcal{A}^{\mathbb{Z}}$ SFT, $h(X_{H,V})$ is computable.*

Proof. Consider that $\widetilde{\mathcal{G}}(H) = \bigsqcup_{i=0}^{p-1} U_i$ is a state-split cycle of length $p > 0$ – the proof for a graph made of several state-split cycles is similar and briefly mentioned at the end. We prove that for any $V \subseteq \mathcal{A}^{\mathbb{Z}}$, $h(X_{H,V})$ is computable. Let V be such an SFT.

Let $\varphi\colon \mathcal{A}^{\mathbb{Z}} \to \{0,\dots,p-1\}^{\mathbb{Z}}$ be the factor that maps any symbol in the component U_i, on i. We also call φ its restricted version to $\mathcal{A}^*$.

For $n \in \mathbb{N}$, $j \in \{0,\dots,p-1\}$ and $u \in \{0,\dots,p-1\}^n$, let $u^j = u + (j,\dots,j)$ with the addition made modulo p. We also define $S_u = \{w \in \mathcal{L}_V(n) \mid \varphi(w) = u\}$ and $N_u = |S_u|$.

We have the following, for integers m and n:

$$N_{X_{H,V}}(pm,n) = \sum_{u \in \{0,\dots,p-1\}^n} \left(\Pi_{j=0}^{p-1} N_{u^j}\right)^m.$$

Indeed, a rectangle of size $pm \times n$ in $X_{H,V}$ is given by a vertical word u in $\{0,\dots,p-1\}^n$ that fixes for the whole rectangle where elements of each U_i will be. Then column j can be made of any succession of n symbols that respects u^j – that are in the correct U_i's.

Therefore, we have:

$$\begin{aligned}
h(X_{H,V}) &= \lim_{n\to+\infty} \frac{\log_2\left(N_{X_{H,V}}(pn,n)\right)}{pn^2} \\
&= \lim_{n\to+\infty} \frac{\log_2\left(\sum_{u\in\{0,\dots,p-1\}^n}\left(\Pi_{j=0}^{p-1} N_{u^j}\right)^n\right)}{pn^2} \\
&= \lim_{n\to+\infty} \frac{\log_2\left(\left(\max\limits_{v\in\{0,\dots,p-1\}^n} \Pi_{j=0}^{p-1} N_{v^j}\right)^n \sum_{u\in\{0,\dots,p-1\}^n} \frac{\left(\Pi_{j=0}^{p-1} N_{u^j}\right)^n}{\left(\max\limits_{v\in\{0,\dots,p-1\}^n} \Pi_{j=0}^{p-1} N_{v^j}\right)^n}\right)}{pn^2} \\
&= \lim_{n\to+\infty} \frac{\log_2\left(\left(\max\limits_{v\in\{0,\dots,p-1\}^n} \Pi_{j=0}^{p-1} N_{v^j}\right)^n\right)}{pn^2} + \lim_{n\to+\infty} \frac{\log_2\left(\sum_{u\in\{0,\dots,p-1\}^n} \frac{\left(\Pi_{j=0}^{p-1} N_{u^j}\right)^n}{\left(\max\limits_{v\in\{0,\dots,p-1\}^n} \Pi_{j=0}^{p-1} N_{v^j}\right)^n}\right)}{pn^2} \\
&= \lim_{n\to+\infty} \frac{\log_2\left(\max\limits_{v\in\{0,\dots,p-1\}^n} \Pi_{j=0}^{p-1} N_{v^j}\right)}{pn}
\end{aligned}$$

since in the penultimate line the second term can be bounded by 0 from below (at least one $u \in \{0,\dots,p-1\}^n$ in the index of the sum reaches the maximum of the denominator, so the sum is at least 1), and by $\frac{\log_2(p^n)}{pn^2}$ from above, which tends to 0.

With the previous computation, it is also clear that for all $n \in \mathbb{N}$,

$$\frac{\log_2\left(N_{X_{H,V}}(pn,n)\right)}{pn^2} \geq \frac{\log_2\left(\max\limits_{v\in\{0,\dots,p-1\}^n} \Pi_{j=0}^{p-1} N_{v^j}\right)}{pn}$$

hence

$$h(X_{H,V}) \geq \lim_{n\to+\infty} \frac{\log_2\left(\max_{v\in\{0,\dots,p-1\}^n} \Pi_{j=0}^{p-1} N_{v^j}\right)}{pn}$$

From this we can deduce that the sequence $\frac{\log_2\left(\max_{v\in\{0,\dots,p-1\}^n} \Pi_{j=0}^{p-1} N_{v^j}\right)}{pn}$ actually converges to $h(X_{H,V})$ from below. There is a Turing Machine, constructed algorithmically with V, that computes any term of it. Indeed, for any $v \in \{0,\dots,p-1\}^n$, N_v can be computed because it depends on a one-dimensional SFT. Then the max on a finite set and all the other operations are also doable. As a consequence, $h(X_{H,V})$ is left-recursively enumerable (i.e. a Σ_1^0 real number). Being, by Theorem 8 from [HM10], right-recursively enumerable, it is computable.

The proof is similar if $\widetilde{\mathcal{G}}(H)$ is made of, say, $k > 1$ state-split cycles C_j numbered from 1 to k: consider p the LCM of their periods; φ projects words of $\mathcal{A}^n$ on $(\{1,\dots,k\}\times\{0,\dots,p-1\})^n$ that indicates for each letter to which C_j it belongs, then to which U_i^j of that C_j. Once again, one vertical word of length n is enough to reconstruct, for any $m \in \mathbb{N}$, a $pm \times n$ rectangle except for the precise choice of an element in each U_i^j. The rest of the computation is similar. □

6.3.3 Condition D, uncomputable entropy

Other horizontal constraints that verify condition D, such as some of reflexive type, allow for a greater range of accessible entropies. In what follows, we show that we can transform most graphs with one SCC that originally *do not* verify condition D into graphs that do – by adding a loop to each and every vertex – yet ensure that the resulting graph is robust enough so that the generic construction in Section 6.1.3 can still be mostly performed. Notably, we keep the capacity to obtain, up to a multiplicative factor, any right-recursively enumerable entropy by choosing adequate interplaying vertical constraints.

Proposition 6.12. *Let $H \subset \mathcal{A}^{\mathbb{Z}}$ be a nearest-neighbor SFT whose Rauzy graph is made of only one strongly connected component and does not respect condition D. We also require the graph to have no vertex at distance one of all others.*

Let $\tilde{H} \subset \mathcal{A}^{\mathbb{Z}}$ be the nearest-neighbor SFT obtained when adding a loop to every vertex of the Rauzy graph of H. Then $\tilde{H}$ verifies condition D, but there exists a subshift of finite type $\tilde{V}$, which can be obtained algorithmically with H as input, such that $h(X_{\tilde{H},\tilde{V}})$ is not computable.

Proof. Without loss of generality, one can consider that the starting SFT H has a Rauzy graph with a loop on every vertex except one. Indeed, such a graph can be obtained algorithmically from any one with less loops, and both yield the same $\tilde{H}$ used to actually perform the construction.

Now, for a given input H that verifies condition C (Definition 6.3), modify the corresponding construction of V performed in Section 6.1.3 with the following additions:

- adding two extra symbols c_i^1 at the bottom of any (i,j)-coding micro-slice;
- and forcing an $(i+1, j+1)$ macro-slice below any (i,j) macro-slice, instead of having each column based on the repetition of a single pair (i,j).

One can check that these two modifications are so that Proposition 6.5 and Proposition 6.6 still hold true. For the remainder of this proof, we now consider Section 6.1.3 to be performed with these modifications.

Consider, for the rest of the proof, a Wang shift W with uncomputable entropy. By the same reasoning as the proof of Proposition 6.9, $h(X_{H,V_W})$ is also uncomputable (as it is equal to $h(W)$ divided by some H-dependent integers). Now, the only thing left to prove is that X_{H,V_W} and $X_{\tilde{H},V_W}$ contain mostly the same configurations, and consequently have the same entropy.

An in-depth analysis of $X_{\tilde{H},V_W}$ comes from checking what fails in performing Proposition 6.5 and Proposition 6.6 with $\widetilde{\mathcal{G}}(\tilde{H})$ *as if it verified condition* C, with the exact same cycles as done with $\widetilde{\mathcal{G}}(H)$. There is only one added edge in $\widetilde{\mathcal{G}}(\tilde{H})$ that may create new configurations in the resulting two-dimensional SFT. The only element at risk in the reasoning is the presence of a 0-length uniform shortcut; notably, everything holds regarding attractive/repulsive vertices due to the hypotheses on $\widetilde{\mathcal{G}}(H)$ here.

What that added 0-length uniform shortcut can cause in $X_{\tilde{H},V_W}$ is to allow an (i,j) macro-slice to be followed by these types of macro-slices:

1. an $(i+1, j+1)$ macro-slice;
2. another (i,j) macro-slice;
3. an $(i+1, j+1)$ macro-slice but the second slice is exactly one cell down;
4. another (i,j) macro-slice but the second slice is exactly one cell up;
5. an $(i+1, j)$ macro-slice;
6. an $(i, j+1)$ macro-slice;
7. an $(i+1, j)$ macro-slice but the second slice is exactly one cell down;
8. an $(i, j+1)$ macro-slice but the second slice is exactly one cell up.

Case 1 is the one that normally happens. Case 2 will be acceptable because it does not bring entropy, as seen below. Cases 5, 6, 7 and 8 collapse on other cases because $\widetilde{\mathcal{G}}(H)$ has a C^2 made of a unique element in its generic construction, causing the j index to be irrelevant – and as we perform the construction with $\widetilde{\mathcal{G}}(\tilde{H})$ *as is*, it holds too. Cases 3 and 4 are the only problematic ones; they are actually solved thanks to the additions we did to V_W in the present proof. Indeed, one-cell shifts between two columns (cases 3 and 4 in the enumeration above) are now forbidden, else the new C^1 elements at the bottom of each coding micro-slice would be in contact with another element of C^1 at distance 2 (due to the new way macro-slices succede to each other vertically), resulting in a uniform shortcut of length 2 in the Rauzy graph of $\tilde{H}$, which is forbidden.

The only part left is to prove that $h(X_{\tilde{H},V_W}) = h(X_{H,V_W})$ in spite of case 2.

Since the only difference between $X_{\tilde{H},V_W}$ and X_{H,V_W} is the possibility to repeat a column several times in a row, we have

$$N_{X_{\tilde{H},V_W}}(n,n) \geq N_{X_{H,V_W}}(n,n)$$

but also

$$\begin{aligned}
N_{X_{\tilde{H},V_W}}(n,n) &= \sum_{k=1}^{n} \sum_{i_\ell | i_1+\cdots+i_k=n} N_{X_{H,V_W}}(i_\ell, n) \\
&\leq \sum_{k=1}^{n} \binom{n+k-1}{n} N_{X_{H,V_W}}(n,n) \\
&= \binom{2n}{n+1} N_{X_{H,V_W}}(n,n)
\end{aligned}$$

by counting, for n columns, how many different types of them there are – and then each i_ℓ counts for a given type how many times it is juxtaposed with itself. The sequence $\log_2(\binom{2n}{n+1})$ has growth rate $\mathcal{O}(2n\log_2(2n))$, and therefore applying $\lim_n \dfrac{\log_2(.)}{n^2}$ to these bounds shows that the entropy is the same. $\square$

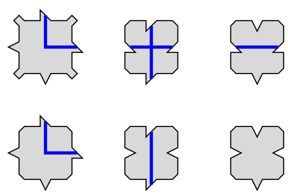
Part III

Subshifts on Finitely Generated Groups

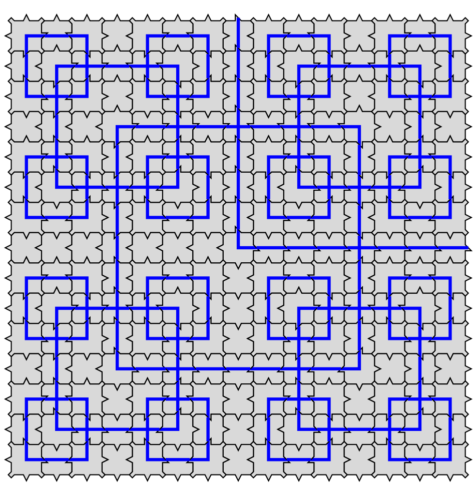
In the previous parts, subshifts have been considered in one ($\mathbb{Z}$) and two ($\mathbb{Z}^2$) dimensions. These two subfields of symbolic dynamics, one born from the discrete encoding of trajectories in continuous dynamical systems [MH38] in articles by Morse and Hedlund in the 40s (see [CN06] for a comprehensive historiography), the other present in the earliest forms of tilings introduced as computational models by Wang in the 60s [Wan61], have attracted vivid interest in the decades that followed. Indeed, the various differences of behavior between one-dimensional and two-dimensional subshifts (range of available entropies, decidability of the Domino Problem) are still under scrutiny in the present day (see Part II of this PhD).

Nevertheless, as mentioned in Section 4.1, several results on $\mathbb{Z}^2$ can be extended in a straightforward manner to any $\mathbb{Z}^d, d > 0$ to obtain d-dimensional subshifts. Several results on $\mathbb{Z}^d, d > 0$ are the same as on $\mathbb{Z}^2$ (entropy [HM10], Domino Problem [ABJ18]). However, some of them differ: consider for instance the notion of periodicity partly mentioned in Section 6.2.1 for two-dimensional subshifts.

It has been known for a long time that if a set of dominoes tiles $\mathbb{Z}$, it is always possible to do it in a periodic fashion. Wang suspected that it was the same for Wang tiles on $\mathbb{Z}^2$. However, a few years later, one of his students, Berger, proved otherwise by providing a set of Wang tiles that tiled the plane but only aperiodically [Ber66]. Numerous aperiodic sets of Wang tiles have been provided by many since then (see for example [Rob71, Kar96, CI96, JR15]). The most well-known may be the Robinson tile set, which contains square of arbitrarily large size that break any possibility for periodicity. Its alphabet is the following, including the symmetries and rotations of the given tiles (omitted for simplification):



An example of pattern from a configuration is as follows – notice the nested blue squares of increasing size that are made to appear.



Still, an important property of SFTs on $\mathbb{Z}^2$ is the following: if there exists a configuration with one direction of periodicity, then there exists another configuration with two directions of periodicity. Intermediate cases arise on $\mathbb{Z}^d, d > 0$: on $\mathbb{Z}^3$, for instance, there are SFTs that contain a configuration with one direction of periodicity, but no configuration with two directions of periodicity or more. To be convinced of this fact, simply consider the subshift made of adding a third dimension to the Robinson subshift, with no additional restriction. A configuration made of the same Robinson tiling repeated at any height will be periodic in the added direction. However, no configuration will have any "planar" periodicity, by two-dimensional aperiodicity of the Robinson subshift. These properties lead to the introduction of two distinct mathematical definitions of aperiodicity of a given subshift, to separate these behaviors. The first is weak aperiodicity, as witnessed in the previous example: not a configuration in the subshift is made of a single repeated finite patch, or equivalently no configuration has finite orbit (that is, a finite number of translates) – but some may have one direction of periodicity. The second is strong aperiodicity: no configuration has any direction of periodicity at all, or equivalently all of them have empty stabilizer (that is, no translation whatsoever leaves any configuration invariant).

Now, a natural question stemming from this first generalization to $\mathbb{Z}^d$ is: can we generalize further to other underlying structures, and study how the entropy, Domino Problem and aperiodicity are affected? Tile sets, the simplest subshifts there are, have so far been made of d-dimensional cubes, based on the d directions of the canonical base of $\mathbb{Z}^d$. Basically, this means coloring an underlying infinite graph where all vertices look locally similar, using a complicated "color" for each vertex, that is actually made of $2d$ colors. Other graphs possess such a structure with local similarity on all vertices, though. Some of them, called *Cayley graphs*, describe the essential algebraic structures that are *groups*.

Trying to extend what has been developed for $\mathbb{Z}^d$ to any finitely generated group (see Definition 7.11) works surprisingly well: after all, the usual discrete space we consider when representing $\mathbb{Z}^d$ is nothing more than the most commonly considered Cayley graph of said $\mathbb{Z}^d$, seen as a group. Consequently, the same can be done on any finitely generated group using symbols decorating vertices of graphs. That being said, be careful: these considerations of graph colorings are a visual help that still depends on the generators chosen for the group. The important fact is that even the more fundamental mathematical structures behind subshifts on groups work well by themselves: after all, the theory of dynamical systems was originally made for group actions in general, as seen in Definition 7.1. As such, building a full shift on any $\mathcal{A}^G$ is straightforward; and so definitions of G-subshifts and G-SFTs for any group G result naturally from this, even outside of any visualization through graph colorings.

$\mathbb{Z}$-subshifts are well-known (see for instance [LM95, HM10]), and even $\mathbb{Z}^d$-subshifts have had interesting developments over the past two decades. Subshifts on any finitely generated group have been attracting more and more attention in symbolic dynamics [Jea15, CGS17, Coh17, ABJ18, ABM19, BMP22], though they have also been thoroughly studied as Bernoulli shifts from a measure-theoretic point of view (see for instance the Ornstein isomorphism theorem, that dates back to 1970). A notable interest for subshifts on groups springs from the apparent links between the Domino Problem [CP15, ABJ18] and the weak and strong notions of aperiodicity: several conjectures have been made to that end, see Section 7.2.2, that make for a yet incomplete classification of groups depending on their behavior. Specifying that classification for a small but meaningful class of groups is the main goal of the current part of this thesis.

Baumslag-Solitar groups of parameters m and n, commonly denoted by $BS(m,n)$, are a specific class of groups that initially gathered interest in symbolic dynamics because of their simple description and rich properties. It is known that their Domino Problem is undecidable [AK13, AK21]; and Aubrun and Kari built a weakly aperiodic SFT in order to prove it. Only the $BS(2,3)$ case is detailed in their original paper; and an explicit period is provided for a given configuration, which shows that the resulting SFT, although weakly aperiodic, is not strongly aperiodic.

In this part of the present thesis, we investigate this result further: after a few definitions (Chapter 7); we thoroughly reintroduce Aubrun and Kari's construction in the general case, and provide a precise proof of its weak aperiodicity by encoding piecewise linear maps (Section 8.1). Then, we show that the resulting SFT is weakly but not strongly aperiodic on any $BS(m,n)$ (Section 8.1.4) (as sketched in [AK13]); except on $BS(1,n)$ where, with some extra work, we prove that it is actually strongly aperiodic (Section 8.1.6). In Section 8.2, using a different technique based on substitutions on words, we exhibit a weakly but not strongly aperiodic SFT on $BS(1,n)$. Finally, by using tools from group theory and a theorem by Jeandel [Jea15], we build a strongly aperiodic SFT on any $BS(n,n)$ in Section 8.3. These new results are summarized in the following table (in bold):

Group	Strongly aperiodic SFT	Weakly-not-strongly aperiodic SFT
$\mathbb{Z}^2 \cong BS(1,1)$	Yes (Berger [Ber66])	No (Folklore)
$BS(1,n)$	**Yes, adapted from Aubrun-Kari** (Section 8.1.6)	**Yes, using substitutions** (Section 8.2)
$BS(n,n)$	**Yes, using a theorem by Jeandel** (Section 8.3)	Yes (Section 8.1.4, Aubrun-Kari [AK13])
$BS(m,n)$	Yes (recent result from [ABHT22])	Yes (Section 8.1.4, Aubrun-Kari [AK13])

The end result of the last part of this PhD is that any residually finite Baumslag-Solitar group $BS(m,n)$ with $|m| \geq 2$ or $|n| \geq 2$ has a strongly aperiodic SFT and a weakly but not strongly aperiodic SFT.

Chapter 7

Subshifts on Groups and Aperiodicity

This chapter provides the necessary notions to understand Chapter 8. The reader may skip parts or jump from one section to another as they see fit. The chapter is broadly divided as follows:

- Section 7.1 defines *subshifts on any finitely generated group G*, in a fashion similar to Section 4.1;
- Section 7.2 is a section dedicated to a conjugacy invariant that takes the center stage in the entirety of Chapter 8: *aperiodicity*. It also precises some context on its ties with the Domino Problem on groups;
- Section 7.3.1 introduces the basics on *presentations of groups* and *Cayley graphs*, which are used to visualize specific subshifts on groups in Section 7.3.2;
- Section 7.4 refines the previous notions for the one class of groups studied in this entire part, the *Baumslag-Solitar groups*. It does so mathematically (Section 7.4.1) then it builds the foundation for specific subshifts on these groups that are visualizable through their Cayley graphs (Section 7.4.2 and Section 7.4.3) called *Cycle shifts*;
- Section 7.5 ends the chapter by introducing a tool from combinatorics on words: *substitutions*. It is of use in a construction of subshifts in Section 8.2.

Throughout this chapter and the next one, we assume basic knowledge about groups as a mathematical structure. To simplify notations, we will not use any symbol to note the law of a given group (except in some special cases such as $\mathbb{Z}$); we will merely write concatenated elements. The identity element will be noted e.

7.1 Subshifts on groups

In the present part, we focus on subshifts where the group action can be any finitely generated group (see Definition 7.11), not just $\mathbb{Z}$ (as in Section 1.2) or $\mathbb{Z}^2$ (as in Section 4.1).

Definition 7.1. Let X be a topological space with τ its topology. Let G be a group and let $\cdot$ denote a group action $G \curvearrowright X$ (that is, a morphism $G \to Aut(X)$).

Then $(X, \tau, \cdot)$ forms a *dynamical system*.

Remark 35. This definition is a generalization of what is done in Chapter 1 with iterations of a function $f\colon X \to X$. In the present part, that iteration can be seen as the straightforward action $\mathbb{Z} \curvearrowright X$ with $k \cdot x = f^k(x)$ for any $k \in \mathbb{Z}$, $x \in X$. We may also speak of dynamical systems with the action of the monoid $\mathbb{N}$, that is, if f is not bijective and the dynamical system nonreversible.

If the topology and the group action are clear, we may denote such a dynamical system as merely X on the group G. This allows for a clear generalization of the notions of full shift and subshift on any finitely generated group:

Definition 7.2. Let $\mathcal{A}$ be a finite alphabet. Let G be a finitely generated group. Let $x \in \mathcal{A}^G$ and $g, h \in G$: G naturally acts on the left on $\mathcal{A}^G$ by

$$(g \cdot x)_h = x_{g^{-1}h}.$$

The set $\mathcal{A}^G$, when endowed with the prodiscrete topology t_π and this action, forms a compact dynamical system $(\mathcal{A}^G, t_\pi, \cdot)$ called the *full shift* on G. We call $x \in \mathcal{A}^G$ a *configuration*.

Remark 36. Writing the action as $(g \cdot x)_h = x_{g^{-1}h}$ is extremely important in nonabelian groups. It also matters in the way patterns are preserved when shifting visual representation of configurations seen through (right) Cayley graphs, as seen in Section 7.3.2.

Definition 7.3. A *pattern* p is an element of some $\mathcal{A}^{P_p}$ where $P_p \subset G$ is finite. We say that a pattern $p \in \mathcal{A}^{P_p}$ *appears* in a configuration $x \in \mathcal{A}^G$ – or that x *contains* p – if there exists $g \in G$ such that for every $h \in P_p$, $(g \cdot x)_h = p_h$, and we write $p \sqsubset x$.

The *subshift* associated to a set of patterns $\mathcal{F}$, called set of *forbidden patterns*, is defined by

$$X_{\mathcal{F}} = \{x \in \mathcal{A}^G \mid \forall p \in \mathcal{F}, p \not\sqsubset x\}$$

that is, $X_{\mathcal{F}}$ is the set of all configurations that do not contain any pattern from $\mathcal{F}$.

As always, a subshift can equivalently be defined as a closed subset of some $\mathcal{A}^G$ under both the topology and the G-action, making it – dynamics-wise – a subsystem of that full shift. It is named, in all generality, a *subshift on G with alphabet $\mathcal{A}$*.

The above definitions are the equivalent on any group of Definition 4.1, Definition 4.3 and Property 4.4.

Remark 37. As in Section 1.2 and Section 4.1, note that there can be several sets of forbidden patterns defining the same subshift X.

Definition 7.4. A *morphism* $h\colon X \to Y$ between two subshifts X and Y on G, with actions $\cdot_X$ and $\cdot_Y$ respectively, is a continuous function so that:

$$\forall g \in G, \forall x \in X, h(g \cdot_X x) = g \cdot_Y h(x).$$

The terms *factor*, *factor map* and *conjugates* mean the same as in Definition 1.19.

The definitions of *SFT*, *sofic* and *effective* subshifts on finitely generated groups are the same as in Definition 1.20. The same implications as in Property 1.22 hold; the reverse implications may hold too, depending on the chosen group.

7.2 Aperiodicity

7.2.1 Definitions

Definition 7.5. Let X be a subshift on a group G and $x \in X$ be a configuration. The *orbit* of x is the set $Orb_G(x) = \{g \cdot x \mid g \in G\} \subset X$ and its *stabilizer* is $Stab_G(x) = \{g \in G \mid g \cdot x = x\} \subset G$.

Definition 7.6. We say that x is a *strongly periodic configuration* if $|Orb_G(x)| < +\infty$, and that x is a *weakly periodic configuration* if $Stab_G(x) \neq \{e\}$.

Simply speaking, a weakly periodic configuration has at least one translate by the group action that is actually the original configuration; a strongly periodic configuration only has a finite number of distinct translates by the group action. Of course, in infinite groups, any strongly periodic configuration is in particular weakly periodic, justifying the terminology.

Definition 7.7. If no configuration in X is strongly periodic and the subshift is nonempty, then the subshift is said to be *weakly aperiodic*. If no configuration in X is weakly periodic and the subshift is nonempty, then the subshift is said to be *strongly aperiodic*.

Note how being weakly and strongly aperiodic are properties preserved by conjugacy.

Remark 38. In the literature, it is commonly considered that an empty subshift is weakly/strongly aperiodic. We prevent it here to make statements more compact and avoid any problem of "zerology" of the sort.

Note how, in this thesis, we will always speak about a periodic *configuration* and an aperiodic *subshift*, to avoid any confusion.

7.2.2 Known results and ties with the Domino Problem

This subsection is, strictly speaking, superfluous for what follows; but it gives some context on the relevance of what is achieved in Chapter 8.

In few words, as briefly mentioned in the introduction to this part, it is possible to extend the Domino Problem to finitely generated groups [ABJ18] and to study its decidability on them. Interestingly enough, the decidability or undecidability can be "inherited" between groups linked by some relations such as commensurability [CP15] (having conjugated subgroups of finite index). Even more striking, there seems to be a correlation overall between the existence of some aperiodic SFT and the undecidability of the Domino Problem on a group, with only few groups (the virtually free groups, see [Pia08, ABJ18]) known so far for breaking that pattern.

See Table 7.1 for a summary of what is known on the subject. As mentioned in the introduction to this part, subshifts on $\mathbb{Z}^3$ and beyond allow the construction of a weakly but not strongly aperiodic SFT; along with a strongly aperiodic SFT by a product of Robinson tiles. In the table, $\mathbb{H}$ designates the discrete Heisenberg group, for which similar constructions can be performed with some additional care [SSU20]. The number of ends of a group, denoted $e(G)$, is an invariant counting the number of connected components of the Cayley graph of G (see Definition 7.12) as bigger and bigger balls centered on the identity are pruned from it. For an infinite group, it is known to belong to the set $\{1, 2, \infty\}$. Results on groups with two ends [Coh17] and the free groups [Pia08, ABJ18] (see Definition 7.8) seem to corroborate still-standing conjectures [CP15, BS18] that they are the only ones in their respective cell of Table 7.1.

Conjecture 2 (see [CP15])**.** *An infinite f.g. group has no weakly aperiodic SFT if and only if it is virtually* $\mathbb{Z}$*.*

Conjecture 3 (see [BS18])**.** *An infinite f.g. group has decidable Domino Problem if and only if it if virtually free.*

Baumslag-Solitar groups have been thought for a while to be candidates as groups with undecidable Domino Problem, the existence of a weakly aperiodic SFT, but the absence of a strongly aperiodic SFT. What we develop in Chapter 8 as published in [EM22], strengthened by a recent article [ABHT22], shows that they are not those desired candidates.

Remark 39. Recent informal discussions on the subject with N. Pytheas Fogg has brought to light that $\mathbb{Z}^2 * \mathbb{Z}$ is such a group, containing $\mathbb{Z}^2$ (undecidable DP) and with more than one end (no strongly aperiodic SFT) that still retains a weakly aperiodic SFT with a mere adaptation of the Robinson tileset.

Aperiodicity / DP	∃ **strongly aperiodic SFT** ∃ **strict. weakly aperiodic SFT**	∃ **strongly aperiodic SFT** ∄ **strict. weakly aperiodic SFT**	∄ **strongly aperiodic SFT** ∃ **weakly aperiodic SFT**	∄ **weakly aperiodic SFT**
Decidable	?	?	Virtually free groups (non-$\mathbb{Z}$)	Virtually $\mathbb{Z}$ (also known as $e(G) = 2$)
Undecidable	$\mathbb{Z}^d, d \geq 3$ $BS(m, n)$ $\mathbb{H}$	$\mathbb{Z}^2$	$\mathbb{Z}^2 * \mathbb{Z}$	?
	$e(G)$ must be 1	$e(G)$ must be 1		

Table 7.1: A few groups for which both the aperiodicity and the Domino Problem are known. The use of "strict. weakly aperiodic SFT" is a wording used to shorten "weakly but not strongly aperiodic SFT". The results on ends of groups are due to [Coh17].

7.3 On group presentations

7.3.1 Group presentations and Cayley graphs

Here, we remind the reader about many things on presentations and Cayley graphs of groups – one can also find such introduction in [Pia08, Chapter 1.1] for instance.

The concept of words written with symbols is a somewhat informal manner of defining a free group, but we provide it that way not to spend too many pages on its formal definition. A reader can find a proper introduction to the subject in the first few pages of [LS01].

Definition 7.8. Let S be a set. The *free group* over S, denoted by $\mathbb{F}_S$, is the group such that its elements are precisely all the finite words which can be written using characters from S and characters denoted as their inverses – inverses which form a set denoted as S^{-1}. The law used is the natural concatenation. The elements of S are called the *generators* of the group.

All elements written this way are considered different unless their equality comes from group axioms (i.e. for any letter l, we can delete all occurrences of ll^{-1} in a word). Said otherwise,

$$\mathbb{F}_S \cong \left(S \cup S^{-1}\right)^* / \{ll^{-1} \mid l \in S \cup S^{-1}\}$$

The *free group of rank* n, where $n \in \mathbb{N}$, is the unique free group (up to isomorphism) with n generators $\{s_1, ..., s_n\}$. It is denoted by $\mathbb{F}_n$.

Remark 40. More generally two free groups using generating sets of the same cardinality are always isomorphic.

Example 20.

- In $\mathbb{F}_2$, let us call the generators a and b. Then e (the empty word), ab, ba, ab^2a^3, b^{42} are all different elements. However, $ba^{-3}a^2ab^{-1}$ is e.
- $\mathbb{F}_1$ is $\mathbb{Z}$. Indeed, if a designates the only generator, the words one can write are precisely $\{a^k \mid k \in \mathbb{Z}\}$ with $a^0 = e$.

Definition 7.9. Let S be a set and consider $\mathbb{F}_S$ with these generators.

Let $R \subseteq \mathbb{F}_S$ and $\mathcal{R}$ be the smallest normal subgroup of $\mathbb{F}_S$ containing R.

We say that a group G has *presentation* $\langle S \mid R\rangle$ if G is isomorphic to $\mathbb{F}_S/\mathcal{R}$. Once again, the elements of S are called the *generators*; the elements of R are called the *relators*.

Proposition 7.10. *Every group G has at least one presentation.*

Proof. If we simply consider $\mathbb{F}_G$, then we have an obvious surjective morphism $\varphi\colon \mathbb{F}_G \to G$. Its kernel K being normal in $\mathbb{F}_G$, we obtain an isomorphism $G \cong \mathbb{F}_G/K$. Hence $G = \langle F_G \mid K\rangle$, although this presentation may be somewhat "ugly". □

Although the previous definition may sound a bit technical at first sight, it is actually quite natural. Indeed, R only precises the most basic relations that give the identity element in the group (from which any other relation can be deduced using the group structure).

Example 21.

- $\langle a \mid \varnothing\rangle \cong \mathbb{F}_1 = \mathbb{Z}$ with a corresponding to its usual generator 1;
- $\langle a, b \mid aba^{-1}b^{-1}\rangle \cong \mathbb{Z}^2$ with a and b corresponding to its usual generators $(0,1)$ and $(1,0)$. With a slight abuse of notations, we may rewrite the presentation as $\langle a, b \mid ab = ba\rangle$ for simplicity;
- $\langle a, b, c \mid aba^{-1}b^{-1}, abc^{-1}\rangle \cong \mathbb{Z}^2$ with the additional generator c corresponding to $(1,1)$;
- $\langle a, b \mid \varnothing\rangle \cong \mathbb{F}_2$ the free group with two generators;
- $\langle a \mid a^n\rangle \cong \mathbb{Z}/n\mathbb{Z}$;
- $\langle r, s \mid r^n, s^2, (rs)^2\rangle \cong D_n$ the dihedral group of order $2n$, with r rotation and s symmetry.

Remark 41. Of course, a group can have many different presentations, not using the same sets of generators and relators. In what follows, we look for some particularly simple presentations.

Definition 7.11. A group G is said to be *finitely generated* (abbreviated f.g.) if it has a presentation $G = \langle S \mid R\rangle$ with S finite.

Definition 7.12. Given a presentation $\langle S \mid R\rangle$ of a group G, its (right) *Cayley graph* is the directed graph $\Gamma_G = (G, \vec{E_\Gamma})$ whose vertices are the elements of G and the directed edges are of the form (g, gs) with $g \in G$ and $s \in S$ a generator of G.

The relators are not explicitly present in the aforementioned definition, but they implicitly correspond to the elementary cycles of the Cayley graph. Often in practice the graph is considered undirected, or alternatively is built using $S \cup S^{-1}$.

Example 22.

- $\langle a \mid \varnothing\rangle$ has for Cayley graph a chain of vertices with in-degree and out-degree 1, all edges labeled with a, forming the usual "biinfinite line" representation of $\mathbb{Z}$;

- Similarly, $\langle a, b \mid aba^{-1}b^{-1}\rangle$ can be the usual grid representation of $\mathbb{Z}^2$ with base $((0,1),(1,0))$; $\langle a, b, c \mid aba^{-1}b^{-1}, abc^{-1}\rangle$ is the usual $\mathbb{Z}^2$ grid with the addition of $(1,1)$ to the base vectors;

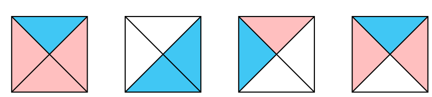
- $\langle a, b \mid \varnothing\rangle$ is the tree with uniform in-degree 2 and out-degree 2.

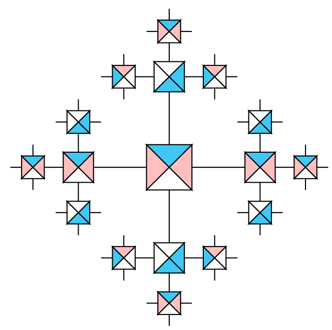
7.3.2 Visualizing configurations

An easy way of building a subshift on a f.g. group G is to consider colorings of one of its Cayley graphs: the alphabet is the set of colors used, and the forbidden patterns are based on adjacency as seen in the Cayley graph. Of course, subshifts on G fundamentally exist without any graphic representation; but those allow for one, given the correct group presentation for G.

Rather notably, one can extend the notion of *Wang shift* to any finitely generated G with a given presentation $\langle S \mid R\rangle$, with $|S| = n$. Characters of the alphabet are $|S \cup S^{-1}|$-tuples $(t_{s_1}, t_{s_1{}^{-1}}, \ldots, t_{s_n}, t_{s_n{}^{-1}})$ – most of the time $2n$-tuples, except if there is some generator s with $s = s^{-1}$. These coordinates precise a color for each "direction" in the Cayley graph, on a given vertex. Forbidden patterns are any nonmatching colors in said Cayley graph.

Example 23. In the picture that follows, we illustrate an alphabet $\mathcal{A}$ made of quadruplets that are Wang tiles adapted to $\mathbb{F}_2$. Then, we represent a local window of a specific configuration from a subshift of $\mathcal{A}^{\mathbb{F}_2}$ that obeys straightforward adjacency rules for the tiles.

$\mathcal{A} = \{$  $\}$



7.4 Baumslag-Solitar groups

7.4.1 Definition

The groups we are interested in this entire part are the *Baumslag-Solitar groups* (abbreviated BS). They are notably defined, using two *nonzero* integers m, n as parameters, by the presentation

$$BS(m,n) = \langle a, b \mid ba^m b^{-1} = a^n \rangle.$$

Remark that $BS(1,1) = \mathbb{Z}^2$, and that $BS(m,n) \cong BS(n,m) \cong BS(-m,-n)$.

The Baumslag-Solitar groups are widely studied, as they are a large class of finitely generated groups – and simple, two-generator one-relator groups – used for numerous counterexamples in group theory. We will not make a extensive list of their properties; nevertheless, we mention one that will only be used here to characterize nicely all the Baumslag-Solitar groups on which we build subshifts in Chapter 8.

Definition 7.13. A group G is said to be *residually finite* if for any $g \in G$ that is not the identity, there is a normal subgroup $N \lhd G$ of finite index such that $g \notin N$.

Proposition 7.14 (Meskin [Mes72])**.** *$BS(m,n)$ is residually finite $\Leftrightarrow$ $|m| = 1$ or $|n| = 1$ or $|m| = |n|$.*

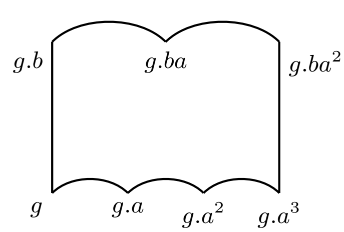
7.4.2 Understanding their usual Cayley Graphs

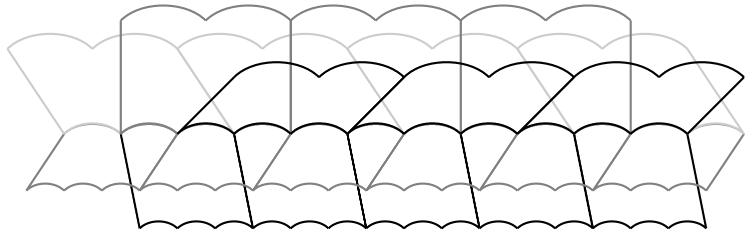
Now, to build subshifts on Baumslag-Solitar groups, we visualize these through their Cayley graph given by the presentation above, and perform a construction that is *close* but not exactly the one of Wang shifts. This has been done previously in [AK13, AK21] by Aubrun and Kari who, adapting a technique by Kari [Kar96], proved the following:

Proposition 7.15. *For all $m, n > 0, BS(m,n)$ admits a weakly aperiodic SFT and has undecidable DP.*

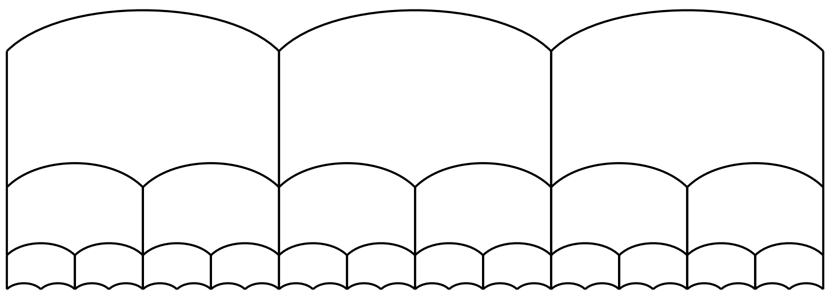
That result is reused and detailed in Section 8.1 of Chapter 8. It is based on Kari's construction on the discrete hyperbolic plane, itself a variant of his construction on $\mathbb{Z}^2$ [Kar96]. The parallel between the discrete hyperbolic plane and the Baumslag-Solitar groups is apparent when considering their Cayley graphs as given by the above presentation.

Cycles in these Cayley graphs, and the local form of said graphs, look like the following – here for $BS(2,3)$.





As can be seen, that Cayley graph is made of lines of successions of a generators (named levels in Chapter 8, see Definition 8.4), and each of these lines has n lines branching out "above" (in the b direction) and m lines branching out "below" (in the b^{-1} direction), connected by rows of cycles. When choosing a succession of lines in the b and b^{-1} directions, one obtains a "sheet" of the Cayley graph, with an appearance that is similar to the following:



Here, this graph corresponds to both a "sheet" of $BS(1,2)$ and the hyperbolic discrete plane. Though the correspondence (and thus, the construction) is slightly less straightforward for any $BS(m,n)$, the similarity is quite clear – it is what gave rise to the article by Aubrun and Kari [AK13] in the first place.

7.4.3 Tiles on BS Groups

A particular class of SFTs is obtained by considering specific colorings of the aforementioned Cayley graphs of Baumslag-Solitar groups, with presentation $\langle a,b \mid ba^m b^{-1} = a^n \rangle$.

A *Cycle shift on a Baumslag-Solitar group* $BS(m,n)$ is a particular SFT where the alphabet is a set of *tiles* τ, which are tuples of colors of the form $s = (t_1^s, \dots, t_m^s, l^s, r^s, b_1^s, \dots, b_n^s)$.

To make notations easier, we denote:

$$\begin{aligned} s(\text{top}_1) &= t_1^s \\ &\vdots \\ s(\text{top}_m) &= t_m^s \\ s(\text{left}) &= l^s \\ s(\text{right}) &= r^s \\ s(\text{bottom}_1) &= b_1^s \\ &\vdots \\ s(\text{bottom}_n) &= b_n^s \end{aligned}$$

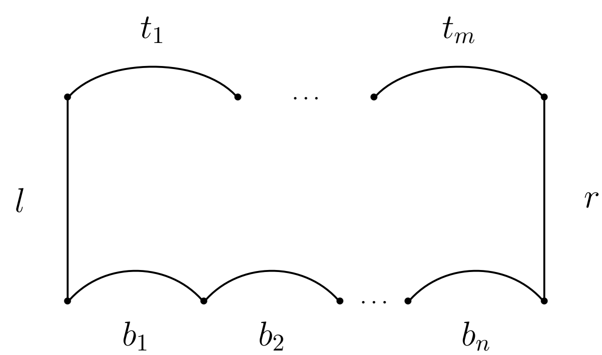


Figure 7.1: A tile of $BS(m,n)$

On the Cayley graph, a tile is considered anchored on a vertex g as the decoration of the cycle of which g is the top left vertex – see Fig. 7.2. We chose the name Cycle shift accordingly – to our knowledge, no standard name exists for these subshifts.

A *tiling* is then a configuration $T \in \tau^{BS(m,n)}$ over the group using the alphabet τ. We say that a tiling is *valid* if the colors of neighboring tiles match – that is, for any $g \in BS(m,n)$ and T_g the associated tile at position g, we must have:

$$\begin{aligned} T_g(\text{right}) &= T_{ga^m}(\text{left}) \\ T_g(\text{top}_k) &= T_{ga^{k-l}b}(\text{bottom}_l) \end{aligned}$$

for any $k \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, n\}$.

See Fig. 7.2 for an illustration of these rules.

Remark 42. The set of all valid tilings for a tileset τ, also refered as the Cycle shift, forms an SFT X^τ, since the tileset gives a finite number of local constraints based on a finite alphabet. In general, it is not necessarily simpler to consider these tilesets – tied to a particular presentation and Cayley graph – instead of a purely mathematical description of the subshift $X \subseteq \mathcal{A}^{BS(m,n)}$; however, as mentioned in Section 7.4.2, in [AK13] the construction heavily uses the visual representation of tiles with numbers on the top and bottom that encode a multiplication by a real number, and Chapter 8 will do the same.

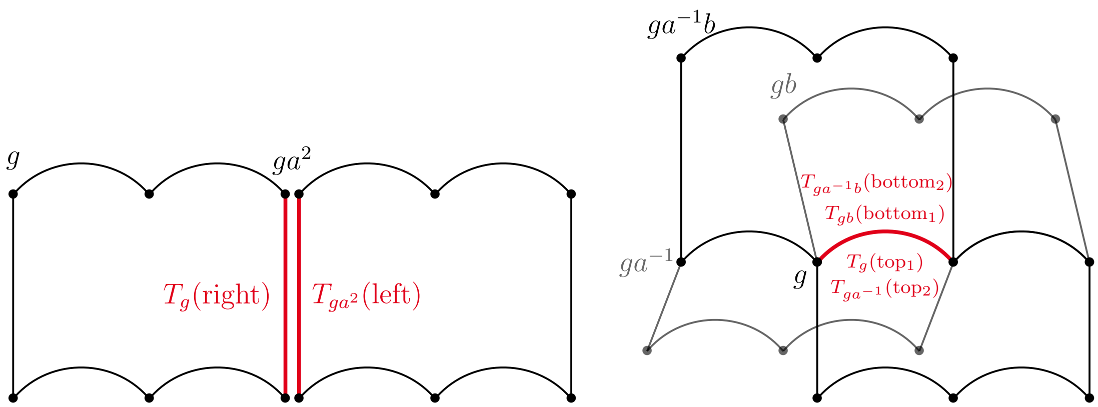


Figure 7.2: The neighbor rules for $BS(2,2)$.

7.5 Substitutions

This section, though less immediately related to subshifts on Baumslag-Solitar groups, is of use in Section 8.2. To explain its use, we need the following definition first.

Definition 7.16. Let $\mathcal{A}^*$ be the set of (finite) words over $\mathcal{A}$. A *substitution* is a map $s\colon \mathcal{A} \to \mathcal{A}^*$. We say it is *uniform* of size $n \in \mathbb{N}$ if for every $a \in \mathcal{A}$, $|s(a)| = n$.

The substitution s can be extended to $\mathcal{A}^*$ by applying it to each letter of the word and concatenating the resulting words. We can also extend s to $\mathcal{A}^{\mathbb{N}_0}$ (resp. $\mathcal{A}^{\mathbb{N}}$) by doing the same, concatenating infinitely many words, the first letter of the first word being at position 0 (resp. 1). Finally, s can be extended to *pointed* biinfinite words, that are words on $\mathcal{A}^{\mathbb{Z}}$ where a point precedes coordinate 0, as illustrated in Fig. 7.3.

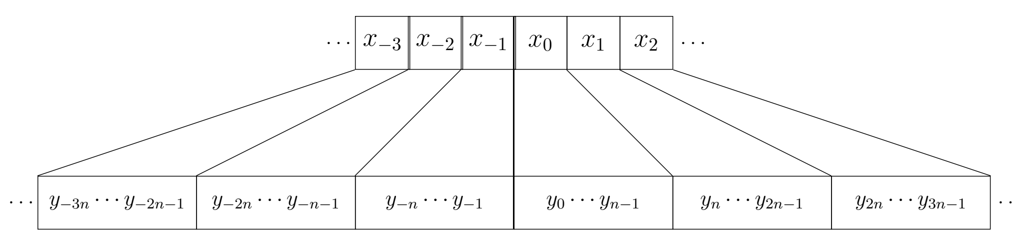


Figure 7.3: How a substitution is applied to a biinfinite word: $y = s(x)$ with s a uniform substitution of size n. Notably, $s(x_{-1}) = y_{-n} \dots y_{-1}$. When writing the word x, a point is added between x_{-1} and x_0, which precises the behavior of the substitution.

The motivation for introducing the mathematical object that are substitutions is simple: the discrete hyperbolic plane can be seen as the *orbit graph* of the substitution $s_0\colon 0 \to 00$ on the trivial alphabet. That substitution duplicates the only existing letter, and its action can consequently be naturally represented as a planar graph with cycles as in $BS(1,2)$ – where any edge above is labeled 0, and any of the two edges below is also labeled 0. Forgetting these labels, we simply obtain the discrete hyperbolic graph. Consequently, any sheet of any $BS(1,n)$ is just the orbit graph of $s_n \colon 0 \to 0^n$. As such, the intuition behind Section 8.2 is that we can try and encode other substitutions of uniform size n in those graphs, possibly obtaining configurations of a subshift that would be weakly aperiodic.

To achieve such a result, we also need some other tools on substitutions.

Definition 7.17. A *fixpoint* of a substitution s is a (possibly biinfinite) word u such that $s(u) = u$.

Definition 7.18. For $a \in \mathcal{A}$ and $u \in \mathcal{A}^*$ with $s(a) = au$, we define the *positive infinite iteration* of s on a:

$$\overrightarrow{s^\omega}(a) = a\ u\ s(u)\ s^2(u) \cdots \in \mathcal{A}^{\mathbb{N}_0}.$$

In the same way, this time for $b \in \mathcal{A}$ and $v \in \mathcal{A}^*$ with $s(b) = vb$, we define the *negative infinite iteration* of s on b:

$$\overleftarrow{{}^\omega s}(b) = \cdots s^2(v)\ s(v)\ v\ b \in \mathcal{A}^{\mathbb{Z}^-}.$$

Notice that, in the previous definition, $s(\overrightarrow{s^\omega}(a)) = \overrightarrow{s^\omega}(a)$ and $s(\overleftarrow{{}^\omega s}(b)) = \overleftarrow{{}^\omega s}(b)$. This leads to the easy creation of pointed biinfinite words that are fixpoints of s: $\overleftarrow{{}^\omega s}(b).\overrightarrow{s^\omega}(a)$ is one. See Lemma 8.23 that makes use of this.

Definition 7.19. For a word u (possibly biinfinite), we define its factor complexity

$$P_u(n) = |\{w \in \mathcal{A}^n \mid w \sqsubset u\}|$$

where $w \sqsubset u$ indicates that w is a subword of u.

Proposition 7.20 (see [CN10])**.** *The factor complexity of a biinfinite word is bounded if and only if that word is periodic, that is, if it is made of the same finite word concatenated infinitely many times.*

Chapter 8

Subshifts on Baumslag-Solitar Groups

The present chapter is divided into three *independent* sections, all of them based on sections from [EM22]:

- Section 8.1 reintroduces a construction from [AK13, AK21], itself based on a construction from [Kar96, CI96]. Its aim and vocabulary are given in Section 8.1.1, Section 8.1.2 to Section 8.1.3 for self-inclusion, before the same conclusion as in [AK13] is reached in Section 8.1.4: in general, that construction yields a weakly aperiodic SFT on $BS(m, n)$ groups. However, a more careful study reveals that the same construction is actually strongly aperiodic on $BS(1, n)$ groups, a result reached in Section 8.1.6.
- Section 8.2 is a brand new approach, based on substitutions as introduced in Section 7.5. It is loosely inspired by [ABM19] and other orbit graphs methods, but directly uses a fitting substitution to tile the Cayley graph of $BS(1, n)$, yielding a weakly but not strongly SFT on it.
- Finally, results from [Jea15] and [CP15] assembled together in Section 8.3, along with the algebraic conjugacy $BS(n, n) \cong (\mathbb{Z} \times \mathbb{F}_n) \rtimes \mathbb{Z}/n\mathbb{Z}$ (re-proved in Proposition 8.29 for self-inclusion), allow for the straightforward proof of the existence of a strongly aperiodic SFT on $BS(n, n)$.

Considering Proposition 7.14; gathering results from Theorem 17, Theorem 18 and Theorem 20; and considering that $BS(-m, -n) \cong BS(-m, n) \cong BS(m, -n) \cong BS(m, n) \cong BS(n, m)$, this section as a whole allows to reach the following result:

Theorem 14. *Residually finite Baumslag-Solitar groups $BS(m, n)$ with $|m| \geq 2$ or $|n| \geq 2$ admit both strongly and weakly-not-strongly aperiodic SFTs.*

An interesting complement to the previous theorem is the recent [ABHT22, Th. 5.9], that shows that all Baumslag-Solitar groups $BS(m, n)$ with $m \neq n$ and $m, n > 1$ admit a strongly aperiodic SFT. The article uses the base technique of Section 8.2, and – roughly speaking – adds a "flow" on the Cayley graph that breaks the possible period from Proposition 8.13 and allows to reach strong aperiodicity.

8.1 On a construction by Aubrun and Kari

In [AK13], Aubrun and Kari provide a weakly aperiodic tileset on Baumslag-Solitar groups, with a proof adapted from [Kar96, CI96]. Their proof focuses on the specific case of $BS(2,3)$, for which they also present a period for one specific configuration, implying that the corresponding tileset is not strongly aperiodic.

A more general version of the construction can be found in [AK21]. We repeat most of it here for the sake of completeness, since we study that construction in more details to obtain additional results: we will show that it yields a weakly but not strongly aperiodic SFT on any $BS(m,n)$ with $|m| \neq 1$ and $|n| \neq 1$, and a strongly aperiodic SFT on any $BS(1,n)$.

8.1.1 Aubrun and Kari's construction

Aubrun and Kari's construction works by encoding orbits of piecewise affine maps applied to real numbers. We will only apply their construction for piecewise linear maps, and begin this section with the necessary definitions.

8.1.2 Definitions

Definition 8.1 (Representation by a sequence)**.** Let $i \in \mathbb{Z}$. We say that a binary biinfinite sequence $(x_k)_{k\in\mathbb{Z}} \in \{i, i+1\}^{\mathbb{Z}}$ *represents* a real number $x \in [i, i+1]$ if there exists an increasing sequence of intervals $I_1 \subset I_2 \subset \cdots \subseteq \mathbb{Z}$ of size at least $1, 2, \ldots$ such that:

$$\lim_{k\to+\infty} \frac{\sum_{j\in I_k} x_j}{|I_k|} = x.$$

Note that if $(x_k)_{k\in\mathbb{Z}}$ is a representation of x, all the shifted sequences $(x_{l+k})_{k\in\mathbb{Z}}$ for every $l \in \mathbb{Z}$ are also representations of x. Note that a sequence $(x_k)_{k\in\mathbb{Z}}$ can *a priori* represent several distinct real numbers since different choices of interval sequences may make it converge to different points. A sequence always represents at least one real number by compactness of $[i, i+1]$.

Then, we define a generalization of piecewise linear maps: multiplicative systems. The main difference with piecewise linear maps is that points may have several images as definition intervals of different pieces might overlap.

Definition 8.2 (Multiplicative system)**.** A *multiplicative system* is a finite set of non-zero linear maps

$$\mathcal{S} = \{f_1 : I_1 \to I'_1, \ldots, f_k : I_k \to I'_k\}$$

with I_i and I'_i closed intervals of $\mathbb{R}$. Its *inverse* is defined to be

$$\mathcal{S}^{-1} = \{f_1^{-1} : I'_1 \to I_1, \ldots, f_k^{-1} : I'_k \to I_k\}.$$

The *image* of $x \in \bigcup_i I_i$ is the set

$$\mathcal{S}(x) = \{y \in \bigcup_j I'_j \mid \exists i, f_i(x) = y\}.$$

The *k-th iteration* of $\mathcal{S}$ on $x \in \bigcup_i I_i$ is then:

$$\begin{cases} \{y \in \mathbb{R} \mid \exists i_1, \ldots, i_k, f_{i_k} \circ \cdots \circ f_{i_1}(x) = y\} & \text{if } k > 0 \\ x & \text{if } k = 0 \\ \{y \in \mathbb{R} \mid \exists i_{-1}, \ldots, i_{-k}, f_{i_{-k}}^{-1} \circ \cdots \circ f_{i_{-1}}^{-1}(x) = y\} & \text{if } k < 0 \end{cases}.$$

Note that if none of the intervals overlap, $\mathcal{S}$ can be represented as a piecewise linear function and the definition of inverse and iteration coincide with the usual ones.

Definition 8.3 (Immortal and periodic points)**.** Let $\mathcal{S} = \{f_1 : I_1 \to I'_1, \ldots, f_k : I_1 \to I'_k\}$ be a multiplicative system. The real number $x \in \mathbb{R}$ is *immortal* if for all $k \in \mathbb{Z}$,

$$\mathcal{S}^k(x) \cap \bigcup_i I_i \neq \varnothing.$$

A *periodic point* for this system is a point $x \in \mathbb{R}$ such that there exists $k \in \mathbb{N}^*$ such that

$$x \in \mathcal{S}^k(x).$$

Definition 8.4 (Level)**.** The *level* of $g \in BS(m,n)$ is the set $\mathcal{L}_g = \{ga^k \mid k \in \mathbb{Z}\}$.

When considering a tiling of $BS(m,n)$, given a line of tiles located between levels $\mathcal{L}_g$ and $\mathcal{L}_{gb^{-1}}$, we talk about the upper side of the line to refer to level $\mathcal{L}_g$, and the lower side of the line to refer to level $\mathcal{L}_{gb^{-1}}$.

Definition 8.5 (Height)**.** The *height* of $g \in BS(m,n)$ is, for any way of writing it as a word in $\{a, b, a^{-1}, b^{-1}\}^*$, its number of b's minus its number of b^{-1}'s; it is denoted as $||g||_b$.

Since the only basic relation in $BS(m,n)$ uses one b and one b^{-1}, all writings of g as a word give the same height. Furthermore, it is actually the height of all elements in its level.

Definition 8.6 (Multiplying tileset)**.** A set of tiles τ *multiplies* by $q \in \mathbb{Q}$ if we have the following equality for any tile $(t_1, \ldots, t_m, l, r, b_1, \ldots, b_n) \in \tau$ (see Fig. 7.1 for the notation):

$$q\frac{t_1 + \cdots + t_m}{m} + l = \frac{b_1 + \cdots + b_n}{n} + r. \tag{8.1}$$

Let τ be a tileset multiplying by $q \in \mathbb{Q}$. If we consider a line of N tiles of τ next to each other without tiling errors (as defined in Section 7.4.3), as left and right colors match, we can average Eq. (8.1):

$$qt + \frac{l}{N} = b + \frac{r}{N}. \tag{8.2}$$

where t is the average of the top labels of the line and b the average of the bottom ones. Therefore, if an infinite line has its upper side representing $x \in \mathbb{R}$ and its lower side representing $y \in \mathbb{R}$, taking the limit of Eq. (8.2) on a well chosen sequence of intervals gives:

$$qx = y.$$

Hence the name of *multiplying tileset* for τ.

8.1.3 A multiplying tileset

Let us define, in a fashion similar to [AK13], a couple of useful functions to build a multiplying tileset. Let $\alpha_{m,n} : \{a, b, a^{-1}, b^{-1}\}^* \to \mathbb{R}$ (or just α when m and n are clear) be defined by the recursion:

$$\begin{cases} \alpha(\varepsilon) = 0 \text{ where } \varepsilon \text{ is the empty word} \\ \alpha(wb) = \alpha(wb^{-1}) = \alpha(w) \\ \alpha(wa) = \alpha(w) + \left(\frac{n}{m}\right)^{||w||_b} \\ \alpha(wa^{-1}) = \alpha(w) - \left(\frac{n}{m}\right)^{||w||_b} . \end{cases}$$

The map α can be extended to elements of $BS(m,n)$, due to the fact that $\alpha(uba^m b^{-1} v) = \alpha(ua^n v)$ for any pair of words u and v in $\{a, b, a^{-1}, b^{-1}\}^*$: $\alpha(g)$ is then $\alpha(w)$ for any word representing g in the group.
Now, we define $\Phi : BS(m,n) \to \mathbb{R}^2$ as follows:

$$\Phi(g) = (\alpha(g), ||g||_b) .$$

The function Φ can be seen as a projection of every element of $BS(m,n)$ on the Euclidean plane $\mathbb{R}^2$.

Finally, let $\lambda : BS(m,n) \to \mathbb{R}$ be defined as

$$\lambda(g) = \frac{1}{m}\left(\frac{m}{n}\right)^{||g||_b} \alpha(g).$$

Let $q \in \mathbb{Q}$ and I an interval, let

$$\begin{aligned} t_j(g,x) &:= \lfloor (m\lambda(g) + j)\, x \rfloor - \lfloor (m\lambda(g) + (j-1))\, x \rfloor \quad \text{for } j = 1 \dots m \\ b_j(g,x) &:= \lfloor (n\lambda(g) + j)\, qx \rfloor - \lfloor (n\lambda(g) + (j-1))\, qx \rfloor \quad \text{for } j = 1 \dots n \\ l(g,x) &:= \frac{1}{m} q \lfloor m\lambda(g) x \rfloor - \frac{1}{n} \lfloor n\lambda(g) qx \rfloor \\ r(g,x) &:= \frac{1}{m} q \lfloor (m\lambda(g) + m)\, x \rfloor - \frac{1}{n} \lfloor (n\lambda(g) + n)\, qx \rfloor \end{aligned}$$

Then, we define the tileset $\tau_{q,I}$ as:

$$\tau_{q,I} = \{(t_1(g,x), \dots, t_m(g,x), l(g,x), r(g,x), b_1(g,x), \dots, b_m(g,x)) \mid g \in BS(m,n), x \in I\}.$$

One can show that Eq. (8.1) holds for these tiles.

Proposition 8.7 (Prop. 6 from [AK21])**.** *Let $k \in \mathbb{Z}$, for any $I \subseteq [k, k+1]$, $\tau_{q,I}$ is a tileset that multiplies by q.*

Let us define a *balanced representation of x*, which is the biinfinite sequence defined for any $z \in \mathbb{R}$ by

$$B_j(x,z) = \lfloor (z+j)x \rfloor - \lfloor (z+j-1)x \rfloor .$$

Note that $B_j(x,z)$ depends on j and z, but can only take two values: $\lfloor x \rfloor$ or $\lfloor x \rfloor + 1$.

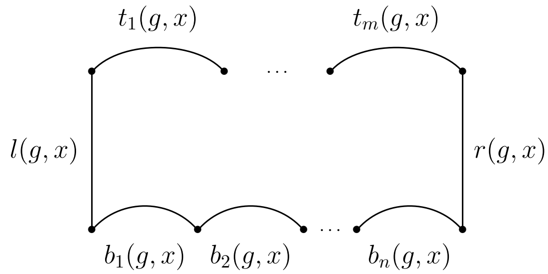


Figure 8.1: One tile from the tileset $\tau_{q,I}$.

Remark 43. If x is irrational and the sequence is made of 0's and 1's, it is called a Sturmian sequence. A binary configuration obtained from a Sturmian sequence is a Sturmian word – they are aperiodic configurations from $\{0,1\}^{\mathbb{Z}}$ that are well known for their numerous definitions (and resulting properties). Indeed, they can be defined from Beatty sequences as mentioned below; from straight lines of irrational slope in the plane; from their factor complexity (Definition 7.19) of $P(n) = n+1$; from the coding of irrational rotations in $\mathbb{S}^1$; and more.

On a related topic, a sequence $(\lfloor jx \rfloor)_{j\in\mathbb{N}}$ by itself (possibly with some offset $r \in [0,1)$, resulting in $\lfloor jx + r \rfloor$) is known as a Beatty sequence.

Proposition 8.8. *Let $q \in \mathbb{Q}, k \in \mathbb{Z}$ and $I \subseteq [k, k+1]$. The upper side of any tile in $\tau_{q,I}$ is of the form*

$$B_j(x, m\lambda(g)), B_{j+1}(x, m\lambda(g)), \ldots, B_{j+m}(x, m\lambda(g));$$

for $x \in I, j \in \mathbb{Z}, g \in BS(m,n)$. In particular, its labels are in $\{k, k+1\}$. The lower side is of the form

$$B_j(qx, m\lambda(gb^{-1})), B_{j+1}(qx, m\lambda(gb^{-1})), \ldots, B_{j+n}(qx, m\lambda(gb^{-1})).$$

Proof. Rewriting the top labels using balanced representation yields

$$B_j(x, m\lambda(g)), B_{j+1}(x, m\lambda(g)), \ldots, B_{j+m}(x, m\lambda(g)).$$

Since each $B_j(x, m\lambda(g))$ is either $\lfloor x \rfloor$ or $\lfloor x \rfloor + 1$, and $x \in [k, k+1)$ (or $x = k+1$, see next sentence), one obtains labels in $\{k, k+1\}$. The case $x = k+1$ has to be treated specifically, by noticing that $B_.(k+1, .) = k+1$ whatever the other values may be. Hence all labels are in $\{k, k+1\}$. For the bottom side, note that $\lambda(gb^{-1}) = \frac{n}{m}\lambda(g)$, which gives labels

$$B_j(qx, m\lambda(gb^{-1})), B_{j+1}(qx, m\lambda(gb^{-1})), \ldots, B_{j+n}(qx, m\lambda(gb^{-1})).$$

□

For our purpose we need to have a finite tileset, because subshifts use a finite alphabet.

Proposition 8.9. *Let $k \in \mathbb{Z}$, for any $I \subseteq [k, k+1]$ the tileset $\tau_{q,I}$ is finite.*

Proof. Proposition 8.8 gives us that there are finitely many top and bottom labels. It remains to prove that there are also finitely many left and right labels.

First of all, one can check that $\lambda(ga^m) = \lambda(g) + 1$, and so $l(ga^m, x) = r(g, x)$. Consequently, we simply have to prove that l lies in a finite set. Let $q = \frac{q_1}{q_2}$ with $q_1, q_2 \in \mathbb{Z}^*$, and write

$$l(g,x) = \frac{nq\lfloor m\lambda(g)x\rfloor - m\lfloor n\lambda(g)qx\rfloor}{mn} = \frac{nq_1\lfloor m\lambda(g)x\rfloor - mq_2\lfloor n\lambda(g)qx\rfloor}{mnq_2}.$$

Since its numerator is an integer bounded by $-nq_1 =: k_1$ from below and $mq_2 =: k_2$ from above using usual inequalities on the floor function, we have that for any $g \in BS(m,n), x \in I$, $mnq_2 l(g,x)$ is an integer comprised between $-nq_1$ and mq_2 (with these values ordered depending on whether q_1 and q_2 are positive or negative). Consequently, for any $g \in BS(m,n), x \in I$, $l(g,x)$ is in a finite set. □

Thanks to this and the multiplying property of $\tau_{q,I}$, we can now use it as a tile set to encode multiplicative systems, in such a way that nonempty tilings correspond to immortal points of the system.

Theorem 15. *Let $\mathcal{S} = \{f_1, \ldots, f_N\}$ be a multiplicative system with*

$$\begin{aligned} f_i : & I_i \to \mathbb{R} \\ & x \mapsto q_i x, \end{aligned}$$

$q_i \in \mathbb{Q}$ and I_i interval with rational bounds included in some $[a_i, a_i+1], a_i \in \mathbb{Z}$. We can explicitly and algorithmically build an SFT $Y_{\mathcal{S}}$ on any $BS(m,n)$ with the following properties:

1. *any top of a line of tiles in a configuration $y \in Y_{\mathcal{S}}$ represents at least one real number $x \in \bigcup_i I_i$.*
2. *if the top of a line of tiles represents a real number $x \in \bigcup_i I_i$, then the bottom of that line represents a real number in $\mathcal{S}(x)$;*
3. *$Y_{\mathcal{S}} \neq \varnothing$ if and only if $\mathcal{S}$ has an immortal point;*

A simplified representation of an element in $Y_{\mathcal{S}}$ is given by Fig. 8.2.

Proof. We build a tileset τ performing the computation by the linear functions $f_i : [a_i, a_i+1] \to \mathbb{R}$; i.e. the linear maps with bigger intervals than the ones defining $\mathcal{S}$. In order to encode the multiplication correctly, one cannot simply take the union of all $\tau_i := \tau_{q_i,[a_i,a_i+1]}$, because tiles coming from different f_i could be mixed on a single line. In order to "synchronize" the computations on every line, we create a product alphabet with the left and right colors of the tiles, and the number of the current function being used. This ensures that one line can have tiles from only one of the $\tau_{q_i,[a_i,a_i+1]}$. Formally,

$$\tau = \left\{(t_1, \ldots, t_m, (l,i), (r,i), b_1, \ldots, b_n) \mid (t_1, \ldots, t_m, l, r, b_1, \ldots, b_n) \in \tau_{q_i,[a_i,a_i+1]}\right\}.$$

The tiling constraints ensure that the left (l_1, i) and right (r_2, i) of two adjacent tiles are the same, hence they come from the same set $\tau_{q_i,[a_i,a_i+1]}$. This way, we can interpret any line of a tiling by τ as being "of color" i for some $i \in \{1, \ldots, N\}$.

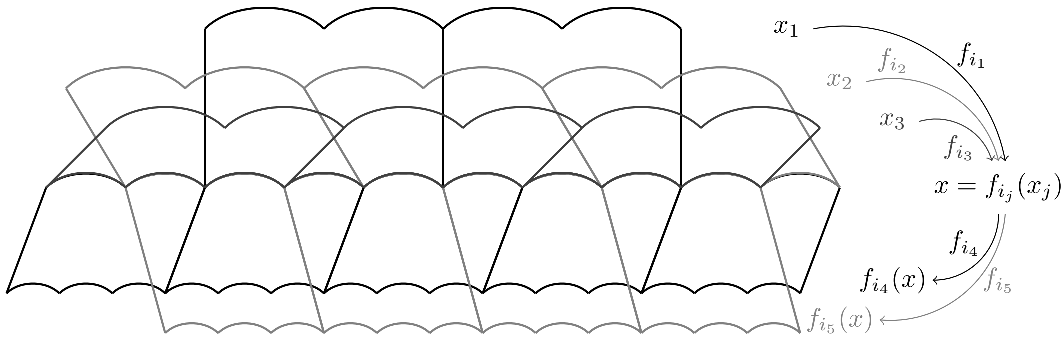


Figure 8.2: Representation of a fragment of a configuration $y \in Y_{\mathcal{S}}$ on $BS(2,3)$. Each x_j is a real number represented by the line next to it in the figure; that line is a biinfinite sequence of digits in some two-element set $\{a_{i_j}, a_{i_j}+1\}$. Each f_{i_j} is a multiplicative map in $\mathcal{S}$; some of them are possibly the same.

Next, we restrict the intervals of real numbers that can be represented in each line of the SFT. Let us write $I_i = [a_i + \frac{d_1^i}{e_1^i}, a_i + 1 - \frac{d_2^i}{e_2^i}]$, for each i. $Y_{\mathcal{S}}$ will be X^τ with the following additional local constraints: on each line of color i, we force the upper side labels to respect

- every e_1^i consecutive labels must contain at least d_1^i labels $a_i + 1$; (8.3)
- every e_2^i consecutive labels must contain at least d_2^i labels a_i. (8.4)

Recall that the top labels of any line of color i only has labels a_i and $a_i + 1$ by definition of τ_i. From this, we can also deduce

- every e_1^i consecutive labels must contain at most $e_1^i - d_1^i$ labels a_i;
- every e_2^i consecutive labels must contain at most $e_2^i - d_2^i$ labels $a_i + 1$.

Since these constraints are on e_1^i or e_2^i consecutive labels, and any other constraint from the τ_i's is on neighboring tiles, $Y_{\mathcal{S}}$ is an SFT.

First, assume that $Y_{\mathcal{S}}$ is nonempty and contains some configuration y. Then, the sequence at the top of each level $\mathcal{L}_{b^k}, k \in \mathbb{Z}$ represents at least one real number x_k, since it is a sequence made of at most two integers. Thanks to the multiplying property of each τ_i (Proposition 8.7), and the fact that the local rules of every line impose that any real number represented belongs to an interval I_i, $(x_k)_{k\in\mathbb{Z}}$ is an infinite orbit of $\mathcal{S}$. Indeed, consider a real number x represented by the upper side of a line of color i. We prove that $x \in I_i = [a + \frac{d_1}{e_1}, a + 1 - \frac{d_2}{e_2}]$ (we drop the i in this paragraph to make notations lighter). Let us write $(J_l)_{l\in\mathbb{N}}$ the intervals from Definition 8.1 (denoted as (I_i) there) on which we compute the mean for the represented real, and let r_l denote the proportion of a's over $(a+1)$'s in the line representing x restricted to J_l. Thanks to the two conditions Eq. (8.3) and Eq. (8.4) (and the two we deduced), we have, for all l in $\mathbb{N}$,

$$\frac{d_2}{e_2 - d_2} \le r_l \le \frac{e_1 - d_1}{d_1}. \tag{8.5}$$

Moreover, since x is the limit of the means computed on each J_l, one can show that

$$x = \lim_{l\to\infty} a + \frac{1}{1+r_l}.$$

Using the left inequality of Eq. (8.5) gives that

$$\frac{1}{1+r_l} \leq \frac{e_2 - d_2}{e_2} = 1 - \frac{d_2}{e_2}.$$

So

$$x \leq a + 1 - \frac{d_2}{e_2}.$$

Similarly, the right inequality of Eq. (8.5) gives

$$x \geq a + \frac{d_1}{e_1},$$

and so $x \in I_i$. This notably ensures Item 1 and, by the use of the τ_i's, Item 2.

Now, assume that S has an immortal point x. Then there exists a sequence $(i_k)_{k\in\mathbb{Z}} \in \{1,\dots,N\}^{\mathbb{Z}}$ such that if we define

$$x_k = \begin{cases} x & \text{if } k = 0 \\ f_{i_{k-1}}(x_{k-1}) = q_{i_{k-1}} x_{k-1} & \text{if } k > 0 \\ f_{i_k}^{-1}(x_{k+1}) = \frac{1}{q_{i_k}} x_{k+1} & \text{if } k < 0 \end{cases},$$

then for all $k \in \mathbb{Z}, x_k \in \bigcup_i I_i$. For every $g \in BS(m,n)$, we place at g a tile from the tileset τ_{i_k} with $k = -||g||_b$, with colors:

$$\begin{aligned} t_j : t_j(g, x_k) &= \lfloor (m\lambda(g) + j)\, x_k \rfloor - \lfloor (m\lambda(g) + (j-1))\, x_k \rfloor \quad \text{for } j = 1 \dots m \\ b_j : b_j(g, x_k) &= \lfloor (n\lambda(g) + j)\, q_{i_k} x_k \rfloor - \lfloor (n\lambda(g) + (j-1))\, q_{i_k} x_k \rfloor \quad \text{for } j = 1 \dots n \\ l : l(g, x_k) &= \left(\frac{1}{m} q_{i_k} \lfloor m\lambda(g) x_k \rfloor - \frac{1}{n} \lfloor n\lambda(g) q_{i_k} x_k \rfloor,\ i_k \right) \\ r : r(g, x_k) &= \left(\frac{1}{m} q_{i_k} \lfloor (m\lambda(g) + m)\, x_k \rfloor - \frac{1}{n} \lfloor (n\lambda(g) + n)\, q_{i_k} x_k \rfloor,\ i_k \right) \end{aligned}$$

These tiles are obviously from the tileset τ. Recall that we have

$$\lambda(ga^m) = \lambda(g) + 1, \tag{8.6}$$

$$\lambda(gb) = \frac{m}{n}\lambda(g).$$

And therefore,

$$\begin{aligned} l(ga^m, x_k) &= \left(\frac{1}{m} q \lfloor m\lambda(ga^m) x_k \rfloor - \frac{1}{n} \lfloor n\lambda(ga^m) q x_k \rfloor,\ i_k \right) \\ &= \left(\frac{1}{m} q_{i_k} \lfloor m(\lambda(g)+1) x_k \rfloor - \frac{1}{n} \lfloor n(\lambda(g)+1) q_{i_k} x_k \rfloor,\ i_k \right) = r(g, x_k), \end{aligned}$$

and for any $j \in \{1,\dots,m\}, p \in \{1,\dots,n\}$,

$$\begin{aligned}
&b_p(ga^{j-p}b, x_{-||ga^{j-p}b||_b})\\
&= \left\lfloor \left(n\lambda(ga^{j-p}b)+p\right) q_{i_{-||ga^{j-p}b||_b}} x_{-||ga^{j-p}b||_b} \right\rfloor - \left\lfloor \left(n\lambda(ga^{j-p}b)+(p-1)\right) q_{i_{-||ga^{j-p}b||_b}} x_{-||ga^{j-p}b||_b} \right\rfloor\\
&= \left\lfloor \left(n\frac{m}{n}\lambda(ga^{j-p})+p\right) q_{i_{-||g||_b-1}} x_{-||g||_b-1} \right\rfloor - \left\lfloor \left(n\frac{m}{n}\lambda(ga^{j-p})+(p-1)\right) q_{i_{-||g||_b-1}} x_{-||g||_b-1} \right\rfloor\\
&= \left\lfloor \left(m(\lambda(g)+\frac{j-p}{m})+p\right) q_{i_{-||g||_b-1}} x_{-||g||_b-1} \right\rfloor - \left\lfloor \left(m(\lambda(g)+\frac{j-p}{m})+(p-1)\right) q_{i_{-||g||_b-1}} x_{-||g||_b-1} \right\rfloor\\
&= \left\lfloor (m\lambda(g)+j)\, q_{i_{-||g||_b-1}} x_{-||g||_b-1} \right\rfloor - \left\lfloor (m\lambda(g)+(j-1))\, q_{i_{-||g||_b-1}} x_{-||g||_b-1} \right\rfloor\\
&= \left\lfloor (m\lambda(g)+j)\, x_{-||g||_b} \right\rfloor - \left\lfloor (m\lambda(g)+(j-1))\, x_{-||g||_b} \right\rfloor\\
&= t_j(g, x_{-||g||_b})
\end{aligned}$$

It remains to show that the labels at the top of every line follow the conditions (8.3) and (8.4). Fix some $g \in BS(m,n)$ and consider the level $\mathcal{L}_g$. Fix $k = -||g||_b$, denote $x := x_k$, $i := i_k$ and drop the i in the other variables to simplify notations. For $j \in \mathbb{Z}$, we write w_j for the label at position j of the considered level, with $t_1(g, x_k)$ being position 1. Remark that

$$w_j = \lfloor (m\lambda(g)+j)\, x \rfloor - \lfloor (m\lambda(g)+(j-1))\, x \rfloor$$

holds for all $j \in \mathbb{Z}$ thanks to Eq. (8.6). Assume that $x \geq a + \frac{d_1}{e_1}$. Let us write N_{j_0} the number of labels a that appear in the word $u = w_{j_0+1} w_{j_0+2} \cdots w_{j_0+e_1}$. If we sum all the labels of u, we have on the one hand

$$\sum_{j=j_0+1}^{j_0+e_1} w_j = N_{j_0} a + (e_1 - N_{j_0})(a+1) = e_1(a+1) - N_{j_0}.$$

And on the other hand,

$$\sum_{j=j_0+1}^{j_0+e_1} w_j = \lfloor (m\lambda(g)+j_0+e_1)\, x \rfloor - \lfloor (m\lambda(g)+j_0)\, x \rfloor.$$

Therefore,

$$e_1(a+1) - N_{j_0} > (m\lambda(g)+j_0+e_1)\, x - 1 - (m\lambda(g)+j_0)\, x = e_1 x - 1 \geq e_1 a + d_1 - 1,$$

which can be rearranged to

$$N_{j_0} \leq e_1 - d_1,$$

which implies (8.3). In the same way, if we assume $x \leq a + 1 - \frac{d_2}{e_2}$, we can show $N_{j_0} \geq d_2$, which is exactly (8.4). This shows Item 3. □

Remark 44. Instead of considering a multiplicative system with rational bounds for the intervals, one can dilate them and encode a dilated system with integer bounds, which is equivalent to the original. For instance, instead of using a linear function f on $[a + \frac{d_1}{e_1}, a + 1 - \frac{d_2}{e_2}]$, one can consider that function f multiplied by $e_1 e_2$, on $[a e_1 e_2 + d_1 e_2, a e_1 e_2 + e_1 e_2 - d_2 e_1]$ which has integer bounds. However, although this would yield a shorter proof, the results would only hold through some sort of conjugation, and we wanted to effectively build a tileset for any multiplicative system.

Note that we only build a multiplying tileset here. Aubrun and Kari provided the details of the encoding of any finite set of affine maps into a tileset of $BS(m,n)$ in [AK21].

8.1.4 A weakly aperiodic SFT on $BS(m,n)$

The SFT $Y_{\mathcal{S}}$ previously defined on a given $BS(m,n)$ is also linked to the periodicity of $\mathcal{S}$.

Theorem 16. *If $\mathcal{S}$ has no periodic point, then $Y_{\mathcal{S}}$ is a weakly aperiodic SFT.*

Proof. We prove the contrapositive.

Assume that $Y_{\mathcal{S}}$ has a strongly periodic configuration y, i.e. $|Orb_{BS(m,n)}(y)| = i$ with $i \in \mathbb{N}$. In particular, each set $\{ga^kg^{-1} \cdot y \mid k \in \mathbb{Z}\}$ is finite of cardinality lesser than i, and therefore for every $g \in BS(m,n)$, there exists $k_g \leq i$ such that $ga^{k_g}g^{-1} \cdot y = y$. If we define $p = i!$, we obtain that for all $g \in BS(m,n)$, $ga^pg^{-1} \cdot y = y$.

Let $g \in BS(m,n)$. Let $h \in \mathcal{L}_g$, i.e. there exists some $k \in \mathbb{Z}$ so that $h = ga^k$. Then

$$\begin{aligned} y_h &= (ga^pg^{-1} \cdot y)_h \\ &= y_{ga^{-p}g^{-1}h} \\ &= y_{ga^{k-p}} \\ &= y_{ha^{-p}} \end{aligned}$$

which means that the level $\mathcal{L}_g$ is p-periodic.

Therefore any level is p-periodic, and consequently represents a unique rational number $\frac{c}{p}$. Since the alphabet of $Y_{\mathcal{S}}$ is finite, there are only finitely many different such rationals. Consequently, there exist two levels $\mathcal{L}_{gb^l}$ and $\mathcal{L}$ with $l > 0$ that represent the same rational number x. By the multiplicative property of $Y_{\mathcal{S}}$,

$$x \in \mathcal{S}^l(x),$$

which means that x is a periodic point for $\mathcal{S}$. □

The consequence of this theorem is that, to obtain a weakly aperiodic SFT on $BS(m,n)$, we only need to explicitly build a multiplicative system with an immortal point but no periodic points. For the rest of this section, notably inspired by [CI96], let

$$\begin{aligned} f_1 \colon [\frac{1}{3}, 1] &\to [\frac{2}{3}, 2] \\ x &\mapsto 2x \end{aligned}$$

$$\begin{aligned} f_2 \colon [1, 2] &\to [\frac{1}{3}, \frac{2}{3}] \\ x &\mapsto \frac{1}{3}x \end{aligned}$$

and

$$\mathcal{S}_0 = \{f_1, f_2\}.$$

It is easy to see that this system has immortal points (in fact, all points of $[\frac{1}{3}, 2]$ are immortal).

Proposition 8.10. *$\mathcal{S}_0$ has immortal points.*

Additionally, since 2 and 3 are relatively prime:

Proposition 8.11. *$\mathcal{S}_0$ has no periodic points.*

Corollary 8.12. *$Y_{\mathcal{S}_0}$ is nonempty and weakly aperiodic.*

However, this construction does not avoid weakly periodic configurations when m and n are not 1, as already remarked by Aubrun and Kari.

Proposition 8.13. *For any $m, n > 1$, $Y_{\mathcal{S}_0}$ on $BS(m,n)$ contains a weakly periodic tiling, with period $p = bab^{-1}aba^{-1}b^{-1}a^{-1}$.*

To prove it, we need the following lemma:

Lemma 8.14. *Let $p = bab^{-1}aba^{-1}b^{-1}a^{-1}$. For any $g \in BS(m,n)$, $\alpha(pg) = \alpha(g)$.*

Proof. Since $||p||_b = 0$, using the definition of α it is easy to show that $\alpha(pg) = \alpha(p) + \alpha(g)$ by recurrence on the length of g. Then, $\alpha(p) = \frac{n}{m} + 1 - \frac{n}{m} - 1 = 0$. □

Proof of Proposition 8.13. We happen to have built a periodic configuration already: the one from the proof of Theorem 15.

Indeed, let x be any real number in $[\frac{1}{3}, 2]$, since they are all immortal for $\mathcal{S}_0$. Then there exists a sequence $(i_k)_{k\in\mathbb{Z}} \in \{1,2\}^{\mathbb{Z}}$ such that if we define

$$x_k = \begin{cases} x & \text{if } k = 0 \\ f_{i_{k-1}}(x_{k-1}) = q_{i_{k-1}} x_{k-1} & \text{if } k > 0 \\ f_{i_k}^{-1}(x_{k+1}) = \frac{1}{q_{i_k}} x_{k+1} & \text{if } k < 0 \end{cases},$$

then for all $k \in \mathbb{Z}, x_k \in [\frac{1}{3}, 2]$. For every $g \in BS(m,n)$, we place at g a tile from the tileset τ_{i_k} with $k = -||g||_b$, with colors:

$$\begin{aligned} t_j : t_j(g, x_k) &= \lfloor (m\lambda(g) + j)\, x_k \rfloor - \lfloor (m\lambda(g) + (j-1))\, x_k \rfloor \quad \text{for } j = 1 \dots m \\ b_j : b_j(g, x_k) &= \lfloor (n\lambda(g) + j)\, q_{i_k} x_k \rfloor - \lfloor (n\lambda(g) + (j-1))\, q_{i_k} x_k \rfloor \quad \text{for } j = 1 \dots n \\ l : l(g, x_k) &= \left(\frac{1}{m} q_{i_k} \lfloor m\lambda(g) x_k \rfloor - \frac{1}{n} \lfloor n\lambda(g) q_{i_k} x_k \rfloor,\ i_k \right) \\ r : r(g, x_k) &= \left(\frac{1}{m} q_{i_k} \lfloor (m\lambda(g) + m)\, x_k \rfloor - \frac{1}{n} \lfloor (n\lambda(g) + n)\, q_{i_k} x_k \rfloor,\ i_k \right) \end{aligned}$$

We already checked that the resulting tiling y was in $Y_{\mathcal{S}_0}$, see the proof of Theorem 15. It remains to show that the tiles at g and at pg are the same for all $g \in BS(m,n)$, to conclude that $p^{-1} \cdot y = y$, or equivalently that p is a period of y. This is actually surprisingly easily, considering that for any $g \in BS(m,n)$, $||pg||_b = ||g||_b$ so they use the same integer k and the same real number x_k; and $\lambda(g) = \lambda(pg)$, see Lemma 8.14. The tiles have the same labels as a consequence of this.

The y consequently defined is p-periodic. The only thing left to show is that p is a nontrivial element of $BS(m,n)$ as long as $m, n > 1$. Since it is freely reduced and does not contain ba^mb^{-1} or $b^{-1}a^nb$ as subwords, and since $BS(m,n)$ is a HNN extension $\mathbb{Z}*_\alpha$ with $\alpha\colon m\mathbb{Z} \to n\mathbb{Z}$, we can apply Britton's Lemma: p cannot be the neutral element.

Therefore $Y_{\mathcal{S}_0}$ contains a configuration with a nontrivial period, and consequently is not strongly aperiodic. □

See Fig. 8.3 for an illustration of a local portion of a configuration in $Y_{\mathcal{S}_0}$.

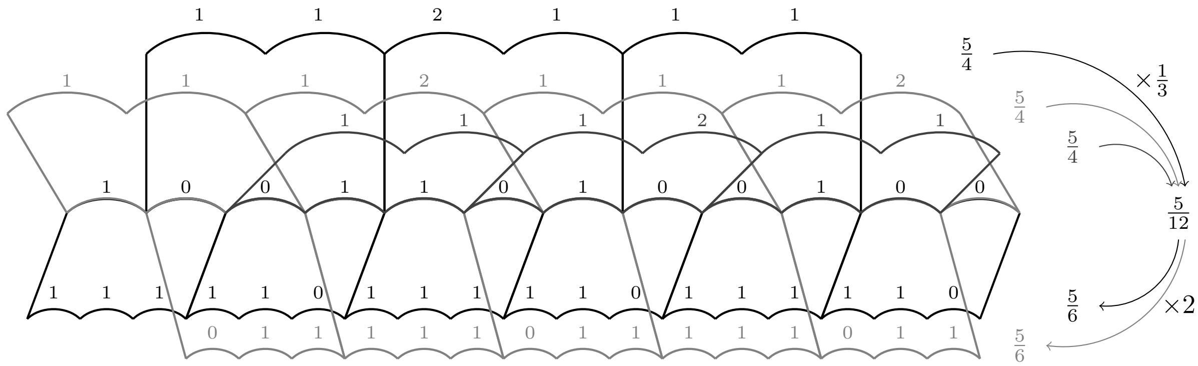


Figure 8.3: Tentative representation of a portion of a configuration in $Y_{\mathcal{S}_0}$ on $BS(2,3)$. Notice how, contrary to the general case from Fig. 8.2, all lines of same height represented here are applied the same multiplicative function.

8.1.5 A deeper understanding of the configurations

We first present additional results on this tileset $Y_{\mathcal{S}_0}$ of $BS(m,n)$. Most of the ideas present in this section were already present in [DGG14] in the context of tilings of the plane.

For a given line $\mathcal{L}_g$, we define the sequence $u_g := (y_{ga^i})_{i\in\mathbb{Z}}$ to be the sequence of digits on the line $\mathcal{L}_g$ (its origin depending on g).

Let f be the following bijective continuous map:

$$f\colon [\tfrac{1}{3},2] \Big/ \tfrac{1}{3}\sim 2 \to [\tfrac{1}{3},2] \Big/ \tfrac{1}{3}\sim 2$$

$$f(x) = \begin{cases} f_1(x) = 2x & \text{if } x \in (\frac{1}{3},1) \\ f_2(x) = \frac{1}{3}x & \text{if } x \in (1,2) \\ \overline{2} & \text{if } x = 1 \\ \frac{2}{3} & \text{if } x = \overline{2} \end{cases}$$

f is strongly related to $\mathcal{S}_0$ because for any $x \in [\frac{1}{3},2]$, for any $k \in \mathbb{Z}$,

$$\mathcal{S}_0{}^k(x) \Big/ \tfrac{1}{3}\sim 2 = f^k(x). \tag{8.7}$$

due to the fact that for our particular $\mathcal{S}_0, \mathcal{S}_0{}^k(x) \subseteq [\frac{1}{3},2]$.

An easy consequence is the following:

Lemma 8.15. *Let $y \in Y_{\mathcal{S}_0}$. Let $g \in BS(m,n)$. Let $x \in [\frac{1}{3},2]$ be a real number represented by u_g; then $u_{gb^{-1}}$ represents $f(x)$ (which means either $\frac{1}{3}$ or 2 if $f(x)=\overline{2}$).*

Proof. u_g represents at least one such x because of Item 1 of Theorem 15. The rest is due to Item 2 of Theorem 15 and Eq. (8.7). □

It turns out that with our choice of multiplicative system $\mathcal{S}_0$, any line can represent only one real number. We need several lemmas to prove this; all inspired by [DGG14].

Lemma 8.16. *Let φ be defined as follows:*

$$\varphi\colon [\tfrac{1}{3},2] \big/ {\tfrac{1}{3} \sim 2} \to [0,1] \big/ {0 \sim 1}$$

$$\varphi(x) = \frac{\log(x)+\log(3)}{\log(2)+\log(3)} \mod 1$$

φ is a well-defined mapping that conjugates the dynamical systems $([\tfrac{1}{3},2] \big/ {\tfrac{1}{3} \sim 2}, t, f)$ and $([0,1] \big/ {0 \sim 1}, t', \varphi\circ f\circ\varphi^{-1})$, where t and t' are the usual topologies on the considered sets.

Proof. Since the action considered on $[0,1] \big/ {0 \sim 1}$ is $\varphi\circ f\circ\varphi^{-1}$, one only needs to check that φ is bijective and continuous to conclude that it yields a conjugation.

φ is clearly continuous everywhere except on $\overline{2}$; there, one can check that the left and right limits both lead to $\varphi(\overline{2}) = 0$ which is correctly defined.

φ^{-1} is defined by $\Phi(y) = \frac{6^y}{3}$ except with collapsed images for 0 and 1. □

Lemma 8.17. *The map $r := \varphi\circ f\circ\varphi^{-1}$ can be considered a rotation of irrational angle $\frac{\log(2)}{\log(2)+\log(3)}$ when identifying $[0,1] \big/ {0 \sim 1}$ and $\mathbb{S}^1$.*

Proof. For every $\alpha \in \varphi((\frac{1}{3},1))$,

$$\begin{aligned}\varphi\circ f\circ\varphi^{-1}(\alpha) &= \varphi(2\varphi^{-1}(\alpha))\\ &= \frac{\log(2)+\log(\varphi^{-1}(\alpha))+\log(3)}{\log(2)+\log(3)} \mod 1\\ &= \alpha + \frac{\log(2)}{\log(2)+\log(3)} \mod 1.\end{aligned}$$

Similarly, for every $\alpha \in \varphi((1,2))$, one has

$$\begin{aligned}\varphi\circ f\circ\varphi^{-1}(\alpha) &= \varphi(\frac{1}{3}\varphi^{-1}(\alpha))\\ &= \frac{\log(\varphi^{-1}(\alpha))}{\log(2)+\log(3)} \mod 1\\ &= \alpha + \frac{\log(2)}{\log(2)+\log(3)} \mod 1.\end{aligned}$$

Finally, $\varphi\circ f\circ\varphi^{-1}(0) = \varphi(f(\overline{2})) = \varphi(\frac{2}{3}) = \frac{\log(2)}{\log(2)+\log(3)}$. □

We now have the tools to prove the following key lemma:

Lemma 8.18. *(Uniqueness of representation) For any $y \in Y_{\mathcal{S}_0}$, for any $g \in BS(m,n)$, the sequence $u_g = (y_{ga^i})_{i\in\mathbb{Z}}$ represents a unique real number.*

Proof. Assume that u_g represents two distinct real numbers x and $z \in [\frac{1}{3}, 2]$. They cannot be $\frac{1}{3}$ and 2 because u_g uses only digits in $\{0,1\}$ or in $\{1,2\}$. Therefore they also are distinct real numbers in $[\frac{1}{3},2]\Big/_{\frac{1}{3}\sim 2}$.

For any $k \in \mathbb{Z}$, notice that $f^k(x) = \varphi^{-1} \circ r^k \circ \varphi(x)$ and same for z, from Lemma 8.16. We will study the behavior of $\varphi(x)$ and $\varphi(z)$ under iterations of r.

The angle $\frac{\log(2)}{\log(2)+\log(3)}$, of which r is a rotation by Lemma 8.17, is irrational. As a consequence, the sets $\{r^k \circ \varphi(x) \mid k \in \mathbb{N}\}$ and $\{r^k \circ \varphi(z) \mid k \in \mathbb{N}\}$ are both dense in $\mathbb{S}^1$.

We introduce $d_{arc}(e^{2i\pi\theta}, e^{2i\pi\psi}) = m(\psi - \theta) \in [0,1)$ for $\theta, \psi \in \mathbb{R}$, where $m(\psi-\theta)$ is the only real number in $[0,1)$ congruent to $\psi - \theta$ mod 1. We call d_{arc} the *oriented arc distance* (measured counterclockwise) between two elements of $\mathbb{S}^1$. It is not a distance *per se* since it is not symmetric and has no triangular inequality, but its basic properties will suffice here. Since r is a rotation, it is easy to check that it preserves d_{arc}. Hence we have that $\forall k \in \mathbb{N}, d_{arc}(r^k \circ \varphi(x), r^k \circ \varphi(z))$ is constant equal to some $c \in [0,1[$. Up to considering $d_{arc}(r^k \circ \varphi(z), r^k \circ \varphi(x))$ instead, and doing the following reasoning by swapping x and z, we can assume that $c \leq \frac{1}{2}$.

Let us split $\mathbb{S}^1$ between $A = \varphi((\frac{1}{3},1))$, $B = \varphi([1,2))$, and $\{\varphi(2)\} = \{\varphi(\frac{1}{3})\} = \{0\}$. We want to show that there is some $l \in \mathbb{N}$ for which $r^l \circ \varphi(x) \in B$ and $r^l \circ \varphi(z) \in A$.

By density of $\{r^k \circ \varphi(x) \mid k \in \mathbb{Z}\}$, there exists some $k_0 \in \mathbb{N}$ such that $d_{arc}(r^{k_0} \circ \varphi(x), 0) < c$ and $r^{k_0} \circ \varphi(x) \in B$. We cannot have $r^{k_0} \circ \varphi(z) = 0$ without contradicting the previous inequality, hence it is either in A or in B. But if it was in B, then the arc from $r^{k_0} \circ \varphi(x)$ to $r^{k_0} \circ \varphi(z)$ would contain all of A. This is not possible because $|A|_{d_{arc}} > \frac{1}{2} \geq c$.

Hence there exists $l = k_0 \in \mathbb{N}$ such that $r^l \circ \varphi(x) \in B$ and $r^l \circ \varphi(z) \in A$.

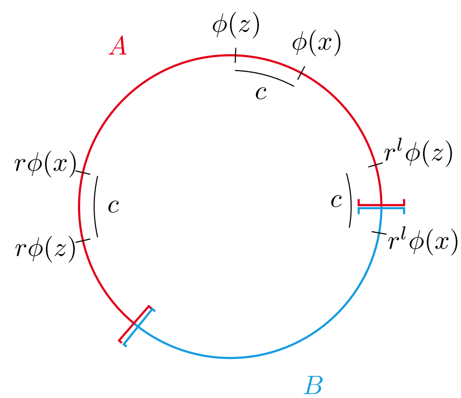


Figure 8.4: Preservation of the oriented arc distance d_{arc} by r and intersection of the arc $\big(r^l \circ \varphi(x), r^l \circ \varphi(z)\big)$ and the boundary between A and B.

Since $r^l = \varphi \circ f^l \circ \varphi^{-1}$, and considering the definitions of A and B, $f^l(z) \in (\frac{1}{3}, 1)$ and $f^l(x) \in [1,2)$. This would cause $f^l(z)$ to be represented by a sequence of 0's and 1's (with an infinite number of 0's) and $f^l(x)$ by a sequence of 1's and 2's. However, the SFT $Y_{\mathcal{S}_0}$ is built such that a line contains only elements in $\{0,1\}$ or $\{1,2\}$, but not both (see proof of Theorem 15): this is a contradiction.

Therefore, x and z must be equal, hence the uniqueness of the real number represented by a given sequence. $\square$

Using previous results, we are now able to prove for $BS(m,n)$ that the real number represented by the sequence u_g only depends on $||g||_b$, its "depth" in the Cayley graph.

Lemma 8.19. *Let $y \in Y_{\mathcal{S}_0}$, and e the identity of $BS(m,n)$. Let x be the unique real number represented by the sequence u_e. Then for every $g \in BS(m,n)$, u_g represents $f^{-||g||_b}(x)$ (with a choice between $\frac{1}{3}$ and 2, possibly different for different g's, if the resulting value is $\overline{2}$).*

Proof. We prove the result by reasoning on words $w \in \{a, b, a^{-1}, b^{-1}\}^*$, by induction on their length. Note that we have no need of proving that different w's representing the same g yield the same result, since this is guaranteed by Lemma 8.18.

The result is true for $g = e$.

Suppose the result is true for words of length $n \in \mathbb{N}_0$. Let w be a word of length n. Then:

- u_{wa} and $u_{wa^{-1}}$ represent the same real number as u_w since they are the same sequence up to an index shift;
- $u_{wb^{-1}}$ represents $f(f^{-||w||_b}(x))$ due to Lemma 8.15 and the induction hypothesis, which is $f^{-||wb^{-1}||_b}(x)$;
- suppose u_{wb} represents y; then u_w represents $f(y)$ due to Lemma 8.15. Then we have, by induction, $y = f^{-||w||_b-1}(x) = f^{-||wb||_b}(x)$.

□

Remark 45. The previous proof heavily relies on the fact that f is a bijection on $[\frac{1}{3}, 2] / \frac{1}{3} \sim 2$, and that we do not have to differentiate between $\frac{1}{3}$ and 2 there.

8.1.6 A strongly aperiodic SFT on $BS(1, n)$

If m or n is equal to 1, then the previous weak period of Proposition 8.13 does not work anymore – it is a trivial element. In fact, we prove in this section that for $BS(1,n)$, $Y_{\mathcal{S}_0}$ is strongly aperiodic.

One key property of $BS(1,n)$ is that there is a simple quasi-normal form for all its elements.

Lemma 8.20. *(Quasi-normal form in $BS(1,n)$) For every $g \in BS(1,n)$, there are integers $k, m \in \mathbb{N}_0$ and $l \in \mathbb{Z}$ such that $g = b^{-k}a^l b^m$.*

Proof. From the definition of $BS(1,n)$, we have that $ba = a^n b$ (1), $ba^{-1} = a^{-n}b$ (2), $ab^{-1} = b^{-1}a^n$ (3) and $a^{-1}b^{-1} = b^{-1}a^{-n}$ (4). Consequently, taking an element of $BS(1,n)$ as a word w written with a and b, we can:

- Move each positive power of b to the right of the word using (1) and (2) repeatedly;
- Move each negative power of b to the left of the word using (3) and (4) repeatedly;

so that we finally get a form for the word w which is: $b^{-k}a^l b^m$ with $k, m \in \mathbb{N}_0$ and $l \in \mathbb{Z}$. □

Remark 46. A general normal form – the same, with k imposed to be minimal – can be obtained from Britton's Lemma. The form obtained here is *not* unique ($a = b^{-1}a^n b$ for instance), but we use it because it admits a simple self-contained proof, and it is enough for what follows: the sum $m - k$ is constant for all writings of a given group element, hence we name it "quasi-normal".

Proof. Indeed, suppose we have $b^{-k}a^l b^m = b^{-k'}a^{l'}b^{m'}$. Then

$$\begin{aligned} b^{-k}a^l &= b^{-k'}a^{l'}b^{-(m-m')} \\ &= b^{-k'-(m-m')}a^{l'n^{m-m'}} \end{aligned}$$

Hence we get $a^{l'n^{m-m'}-l} = b^{-k+k'+m-m'}$. Since it is clear that $a^i = b^j$ if and only if $i = j = 0$ in $BS(1,n)$, we obtain $m - k - (m' - k') = 0$ which is what we wanted. □

This quasi-normal form is the only thing that was missing to prove the following.

Theorem 17. *For every $n \geq 2$, the Baumslag-Solitar group $BS(1,n)$ admits a strongly aperiodic SFT.*

Proof. Let $y \in Y_{\mathcal{S}_0}$, and $g \in Stab_{BS(1,n)}(y)$. Using Lemma 8.20, we can write $g = b^{-k}a^l b^m$ with $k, m \in \mathbb{N}_0, l \in \mathbb{Z}$.

Let x be the real number represented by u_e. By Lemma 8.19, u_g represents $f^{k-m}(x)$. Since $g \in Stab_{BS(1,n)}(y)$, $u_g = u_e$ and so $f^{k-m}(x) = x$ by the uniqueness of the representation from Lemma 8.18. The aperiodicity of f then implies that $k = m$.

Let us assume $l \neq 0$. Then $g = b^{-k}a^l b^k$ and $g^n = b^{-k}(a^n)^l b^k$. We can reduce g^n to $b^{-k+1}a^l b^{k-1}$ using the relation $a^n = bab^{-1}$. More generally, we notice that for any positive integer i, iterating the process i times, we obtain that $g^{n^i} = b^{-k+i}a^l b^{k-i} \in Stab_{BS(1,n)}(y)$.

Since for all i, $g^{n^i} \in Stab_{BS(1,n)}(y)$, we can obtain a contradiction with an argument similar to Prop 6. of [AK13]. We have $b^j a^l b^{-j} \in Stab_{BS(1,n)}(y)$ for any $j \geq -k$. This means that $u_{b^j} = u_{b^j a^l}$ hence u_{b^j} is a l-periodic sequence. We have a finite number of said sequences, since they can only use digits among $\{0, 1, 2\}$. Consequently, there are $j_1 \neq j_2$ such that the two levels $\mathcal{L}_{b^{j_1}}$ and $\mathcal{L}_{b^{j_2}}$ read the same sequence (up to index translation). These two levels represent respectively $f^{j_1}(x)$ and $f^{j_2}(x)$ due to Lemma 8.19, and since the two sequences on these levels are the same, $f^{j_1}(x) = f^{j_2}(x)$. This equality contradicts the fact that f has no periodic point, since we had $j_1 \neq j_2$.

As a consequence, any non-trivial $g \in BS(1,n)$ cannot be in $Stab_{BS(1,n)}(x)$, and we finally get that $Stab_{BS(1,n)}(x) = \{e\}$: $Y_{\mathcal{S}_0}$ is strongly aperiodic. □

Following Theorem 17, a question remains: is the strong aperiodicity of Aubrun and Kari's SFT a property of the group $BS(1,n)$ itself, or does it only arise on carefully chosen SFTs, as $Y_{\mathcal{S}_0}$? Is this because $BS(1,n)$ behaves like $\mathbb{Z}^2$ and all its weakly aperiodic SFTs are also strongly aperiodic, or does Aubrun and Kari's construction happen to be "too much aperiodic"? It turns out that the latter is the correct answer, as we build in the following section an SFT on $BS(1,n)$ that is weakly but not strongly aperiodic.

8.2 A weakly but not strongly aperiodic SFT on $BS(1,n)$

Our weakly but not strongly aperiodic SFT will work by encoding specific substitutions into $BS(1,n)$. Indeed, the Cayley graph of $BS(1,n)$ is very similar to orbit graphs of uniform substitutions (see for example [CGS17, ABM19] for a definition of orbit graphs and another example of a Cayley graph similar to an orbit graph). In this section, we find a set of substitutions that are easy to encode in $BS(1,n)$ (Section 8.2.1), and show how to do it (Section 8.2.2).

8.2.1 The substitutions σ_i

Let $\mathcal{A} = \{0,1\}$. For $r \in \{0,\dots,n-1\}$, let $\sigma_r : \mathcal{A} \to \mathcal{A}^n$ be the following substitution:

$$\sigma_r : \begin{cases} 0 \mapsto 0^{n-r-1}10^r \\ 1 \mapsto 0^n \end{cases} .$$

We may also write $\sigma := \sigma_0$ and call the other ones the *shifts* of σ. Be careful, *in what follows σ designates a substitution and not the shift map*, whose notation is changed to ρ!
Note that, for $c \in \{0,1\}$ and $i \in \{0,\dots,n-1\}$, $\sigma_r(c)_i = 1$ if and only if $c = 0$ and $i = n-r-1$ (starting to count from 0 the indices of the word $\sigma_r(c)$).
All $\sigma_r(0)$ are cyclic permutations of the same finite word. Denote ρ the *shift action* on a biinfinite word u, i.e. $\rho^j(u)_i = u_{i+j}$, as a way to write the action of $\mathbb{Z}$ on $\{0,1\}^{\mathbb{Z}}$.

Lemma 8.21. *For any biinfinite word $u \in \mathcal{A}^{\mathbb{Z}}$, any $i, r \in \{0,\dots,n-1\}$ and $j \in \mathbb{Z}$,*

$$(\sigma_r \circ \rho^j(u))_i = \sigma_r(u_j)_i = (\sigma_r(u))_{nj+i}.$$

Proof. For $i \in \{0,\dots,n-1\}$, $\sigma_r(\rho^j(u))_i$ depends on the letter of $\rho^j(u)$ at position 0 only, that is u_j (See Fig. 8.5), hence $\sigma_r(\rho^j(u))_i = \sigma_r(\rho^j(u)_0)_i = \sigma_r(u_j)_i$.

Similarly, the letter $(\sigma_r(u))_{nj+i}$ does not depend on the totality of u but only on u_j: it is the ith letter of $\sigma_r(u_j)$. □

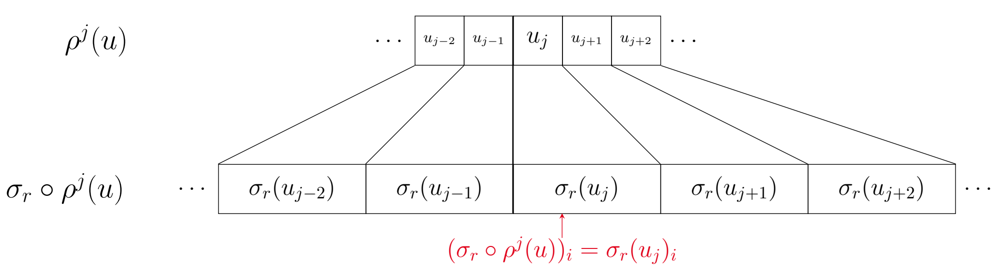


Figure 8.5: Illustration of Lemma 8.21.

Lemma 8.22. *For any $r \in \{0,\dots,n-1\}$,*

$$\sigma_r = \rho^r \circ \sigma.$$

Proof. Let $u \in \mathcal{A}^{\mathbb{Z}}$. Let $i, r \in \{0,\dots,n-1\}$ and $j \in \mathbb{Z}$.

$$\sigma(u)_{nj+i} = \begin{cases} 0 & \text{if } i \neq n-1 \\ \sigma_r(u)_{nj+i-r} & \text{if } i = n-1 \end{cases}$$

Considering that if $i \neq n-1$, $\sigma_r(u)_{nj+i-r} = 0$, we conclude that we always have $\sigma(u)_{nj+i} = \sigma_r(u)_{nj+i-r}$, and so $\sigma_r = \rho^r \circ \sigma$. □

Lemma 8.23. *For $n \geq 3$, σ_1 has a unique fixpoint. For $n = 2$, σ_1 has no fixpoint but ${\sigma_1}^2$ has exactly two fixpoints.*

Proof. Proposition 4 from [SW02] characterizes biinfinite fixpoints of substitutions. In the present case of σ_1, [SW02] states that $w = \sigma_1(w)$ if and only if $w = y.x$ with $x = \overrightarrow{\sigma_1{}^{\omega}}(c)$ and $y = \overleftarrow{{}^{\omega}\sigma_1}(c')$ with $\sigma_1(c) = cv$ and $\sigma_1(c') = uc'$, $u, v \in \{0,1\}^*$, $c, c' \in \{0,1\}$. Notice that $\sigma_1(0) = 0^{n-2}10$ and $\sigma_1(1) = 0^n$, for $n \geq 3$, so the only choice for c and c' is $c = c' = 0$. Then σ_1 has a fixpoint that is $\overleftarrow{{}^{\omega}\sigma_1}(0).\overrightarrow{\sigma_1{}^{\omega}}(0)$ and which is unique.

For $n = 2$ the same reasoning concludes that σ_1 has no fixpoint. However, since ${\sigma_1}^2(0) = 0010$ and ${\sigma_1}^2(1) = 1010$, the same reasoning also yields that ${\sigma_1}^2$ has exactly two fixpoints that are $\overleftarrow{{}^{\omega}({\sigma_1}^2)}(0).\overrightarrow{({\sigma_1}^2)^{\omega}}(0)$ and $\overleftarrow{{}^{\omega}({\sigma_1}^2)}(0).\overrightarrow{({\sigma_1}^2)^{\omega}}(1)$. ☐

Lemma 8.24. *For every $k \in \mathbb{N}$ and every $i_1, \ldots, i_k \in \{0, \ldots, n-1\}$, the fixpoints of $s = \sigma_{i_k} \circ \cdots \circ \sigma_{i_1}$ are aperiodic.*

Proof. To prove the aperiodicity of a fixpoint w of s (in the case where such a fixpoint exists), we follow a proof from [Pan86], simplified for our specific case.

First, let us show that the two subwords 00 and 01 can be found in w.

- For 00, let us define $s' = \sigma_{i_{k-1}} \circ \cdots \circ \sigma_{i_1}$. Then, by definition, $w = \sigma_{i_k}(s'(w))$ (by convention $s'(w) = w$ if $k = 1$). We are going to prove that $s'(w)$ always contains a 1. As a consequence, $w = \sigma_{i_k}(s'(w))$ contains 00 because $\sigma_{i_k}(1) = 0^n$. Suppose $s'(w) = \ldots 000 \ldots$. If $k = 1$, it means that $w = \ldots 000 \ldots$, but then $s(w) \neq w$ so this is impossible. If $k = 2$, then $s' = \sigma_{i_1}$ so the only way to have $s'(w) = \ldots 000 \ldots$ is to have $w = \ldots 111 \ldots$, but again $s(w) \neq w$. If $k \geq 3$, let us define $t = \sigma_{i_{k-3}} \circ \cdots \circ \sigma_{i_1}$. With this notation, $w = \sigma_{i_k} \circ \sigma_{i_{k-1}} \circ \sigma_{i_{k-2}}(t(w))$. The assumption $s'(w) = \ldots 000 \ldots$ causes $\sigma_{i_{k-2}}(t(w)) = \ldots 111 \ldots$. However, this is impossible since $\ldots 111 \ldots$ has no antecedent by $\sigma_{i_k - 2}$. Therefore $s'(w)$ must contain a 1 and we can find 00 in w.

- For 01, the only way for w not to contain 01 is to be of the form $w = \ldots 000 \ldots$, $w = \ldots 111 \ldots$ or $w = \ldots 1100 \ldots$. But it is clear that $s(\ldots 000 \ldots) \neq \ldots 000 \ldots$, $s(\ldots 111 \ldots) \neq \ldots 111 \ldots$ and $s(\ldots 1100 \ldots) \neq \ldots 1100 \ldots$ hence none of them can be fixpoints.

Hence $s(00)$ and $s(01)$ can also be found in w since $s(w) = w$. From this, we build by induction infinitely many words with two possible right extensions. We have $s(00) \neq s(01)$; consider the largest prefix on which they agree, call it u_2, with $|u_2| > 1$. Then both $u_2 0$ and $u_2 1$ can be found in w. Hence $s(u_2 0)$ and $s(u_2 1)$ can also be found in w. We have $s(u_2 0) \neq s(u_2 1)$; consider the largest prefix on which they agree, call it u_3, with $|u_3| > |u_2|$. Then both $u_3 0$ and $u_3 1$ can be found in w. Hence $s(u_3 0)$ and $s(u_3 1)$ can also be found in w.

By induction, we can build subwords of w as large as we want that have two choices for their last letter. Hence the factor complexity of w is unbounded, and so w is aperiodic (see Section 7.5). ☐

8.2.2 Encoding substitutions in $BS(1, n)$

We now show how to encode such substitutions in SFTs of the group $BS(1, n)$ given by a tileset. We define the tileset τ_σ on $BS(1, n), n \in \mathbb{N}, n \geq 2$, to be the set of tiles shown on Fig. 8.6 for all $c \in \{0, 1\}$ and $i \in \{0, \ldots, n-1\}$. Remark that a tile is uniquely defined by the couple (c, i).

This tileset will be the weakly but not strongly aperiodic tileset we are looking for. Lemmas 8.23 and 8.24 study the words that can appear on levels $\mathcal{L}_g$ of the tiling, by looking at the fixpoints of σ_1. They prove that no biinfinite word can be both a fixpoint for the σ_i's and periodic. This

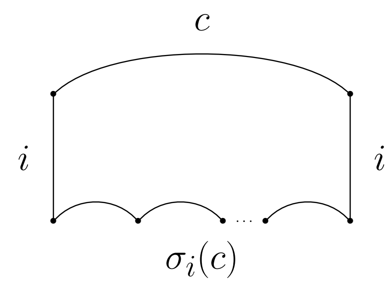


Figure 8.6: Tiles of τ_σ: left and right colors are identical and equal to i, top color is c and bottom colors are equal to $\sigma_i(c)_0, \dots, \sigma_i(c)_{n-1}$.

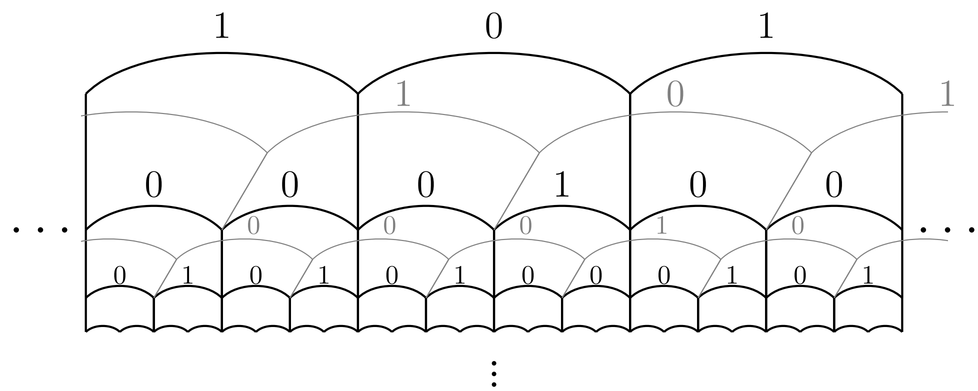


Figure 8.7: Portion of a configuration in X^{τ_σ} on $BS(1,2)$. Notice how σ_0 is applied to any line in black to "go down", whereas lines in gray use σ_1. Several of the branches of the Cayley graph are not illustrated to keep the figure readable.

consequently forbids one direction of periodicity for any configuration we will encode with our tileset, and naturally leads to the following proposition:

Proposition 8.25. *No configuration of X^{τ_σ} can be a^k-periodic for any $k \in \mathbb{N}$.*

Proof. Suppose that there is a configuration x of X^{τ_σ} such that for any $g \in BS(1,n)$, $x_{a^k \cdot g} = x_g$ (a^k-periodicity). Call $w := (x_{a^j})_{j\in\mathbb{Z}}$ the biinfinite word based on level $\mathcal{L}_e$. w is k-periodic by a^k-periodicity of the configuration x. But w is also nk-periodic. Hence $(x_{ba^j})_{j\in\mathbb{Z}}$ is k-periodic. Indeed, by construction, when applying the correct substitution σ_i to x_{ba^j} and $x_{ba^{j+k}}$, one obtains the words $x_{a^{nj}} \dots x_{a^{nj+n-1}}$ and $x_{a^{nj+nk}} \dots x_{a^{nj+nk+n-1}}$ which are one and the same by nk-periodicity of w. Since there is only one preimage for a word by σ_i, $x_{ba^j} = x_{ba^{j+k}}$. By the same argument, one can show that for any integer $l > 0$, $(x_{b^l a^j})_{j\in\mathbb{Z}}$ must be k-periodic. However, these biinfinite sequences only use digits among $\{0,1,2\}$ so there is a finite number of such sequences. In particular, two of these sequences are the same. Since one is obtained from the other by applying the correct succession of σ_i's, we get a periodic sequence that is a fixpoint of some $s = \sigma_{i_N} \circ \cdots \circ \sigma_{i_1}$ for some $i_1, \dots, i_N \in \{0, \dots, n-1\}$. This contradicts Lemma 8.24. ☐

Lemma 8.26. *There exists a weakly periodic configuration in X^{τ_σ} for $n \geq 3$.*

Proof. We define w the unique fixpoint of σ_1 obtained thanks to Lemma 8.23.

Let $f(k) = \lfloor \frac{k}{n} \rfloor$ be the function that maps k to the quotient in the Euclidean division of k by n and $r(k)$ its remainder. We also define $F(k) = f(k+1)$ and $R(k) = r(k+1)$. This means that $nF(k) + R(k) = k+1$, $F(ln) = l + \lfloor \frac{1}{n} \rfloor = l$, but also $F(k+n) = \lfloor \frac{k+1+n}{n} \rfloor = F(k) + 1$, and consequently $F^m(k + n^m) = F^m(k) + 1$.

X^{τ_σ} is nonempty

We define a configuration x describing which tile (c_g, i_g) (a tile being uniquely defined by such a couple) is assigned to g, i.e. $x_g = (c_g, i_g)$, using the quasi-normal form $g = b^{-k}a^l b^m$. Then, we check that x does verify the adjacency rules. Define $x \in {\tau_\sigma}^{BS(1,n)}$ by

$$\begin{cases} x_{b^{-k}a^l} := (w_l, 1) \\ x_{b^{-k}a^l b^m} := (w_{F^m(l)}, R \circ F^{m-1}(l)) \text{ for } m > 0. \end{cases}$$

Remember that Lemma 8.20 states that any $g \in BS(1,n)$ can be written $b^{-k_1}a^{l_1}b^{m_1}$. Suppose it has a second form $b^{-k_2}a^{l_2}b^{m_2}$ with $k_2 > k_1$ up to exchanging the notations (were they equal, it is easy to prove the two forms would be the same). Then $b^{k_1-k_2}a^{l_2}b^{m_2-m_1} = a^{l_1}$, that is, $b^{k_1-k_2}a^{l_2-l_1n^{k_2-k_1}}b^{m_2-m_1} = e$. This means that $k_2 - k_1 = m_2 - m_1 > 0$ and $l_2 = l_1 n^{k_2-k_1}$. With that, we prove our x is well-defined. $k_2 > k_1$ causes $m_2 > 0$ in order to have $k_2 - k_1 = m_2 - m_1 > 0$. Consequently,

$$\begin{aligned} x_{b^{-k_2}a^{l_2}b^{m_2}} &= (w_{F^{m_2}(l_2)}, R \circ F^{m_2-1}(l_2)) \\ &= (w_{F^{m_1+k_2-k_1}(l_1 n^{k_2-k_1})}, R \circ F^{m_1+k_2-k_1-1}(l_1 n^{k_2-k_1})) \\ &= (w_{F^{m_1}(l_1)}, R \circ F^{m_1-1}(l_1)) \\ &= x_{b^{-k_1}a^{l_1}b^{m_1}} \end{aligned}$$

with a variation on the second to last line if $m_1 = 0$: we have $R \circ F^{k_2-k_1-1}(l_1 n^{k_2-k_1}) = R(l_1 n) = 1$.

Now, we prove that $x \in X^{\tau_\sigma}$. Let $g = b^{-k}a^l b^m$.

- If $m > 0$, we have

$$\begin{aligned} x_{ga}(\text{left}) &= x_{b^{-k}a^{l+n^m}b^m}(\text{left}) \\ &= R \circ F^{m-1}(l + n^m) \\ &= R(F^{m-1}(l) + n) \\ &= R \circ F^{m-1}(l) \\ &= x_g(\text{right}). \end{aligned}$$

- If $m = 0$, we have

$$\begin{aligned} x_{ga}(\text{left}) &= x_{b^{-k}a^{l+1}}(\text{left}) \\ &= 1 \\ &= x_g(\text{right}). \end{aligned}$$

Let $j \in \{0,\dots,n-1\}$. We have

$$\begin{aligned}
x_{ga^{-j}b}(\text{bottom}_{j+1}) &= x_{b^{-k}a^{l-jn^m}b^{m+1}}(\text{bottom}_{j+1}) && (g = b^{-k}a^lb^m)\\
&= \sigma_{R\circ F^m(l-jn^m)}(w_{F^{m+1}(l-jn^m)})_j && \text{(by definition of } x)\\
&= \sigma_{R\circ F^m(l-jn^m)}(w)_{nF^{m+1}(l-jn^m)+j} && (\textit{Lemma } 8.21)\\
&= \sigma(w)_{nF^{m+1}(l-jn^m)+j+R\circ F^m(l-jn^m)} && (\textit{Lemma } 8.22)\\
&= \sigma(w)_{F^m(l-jn^m)+j+1} && \text{(by definition of } F \text{ and } R)\\
&= \sigma_1(w)_{F^m(l-jn^m)+j} && (\textit{Lemma } 8.22)\\
&= w_{F^m(l-jn^m)+j} && \text{(since } w \text{ is a fixpoint of } \sigma_1)\\
&= w_{F^m(l)} && (F^m(l-jn^m) = F^m(l)-j)\\
&= x_g(\text{top})
\end{aligned}$$

Consequently, x describes a valid configuration of X^{τ_σ}: all adjacency conditions are verified.

x is b-periodic

With the definition of x, it is easy to check that for any $g \in BS(1,n)$, $x_{bg} = x_g$. Hence it is a weakly periodic configuration. □

We can now obtain our last main theorem of this section:

Theorem 18. *For any $n \in \mathbb{N}, n \geq 2$, the tileset τ_σ forms a weakly aperiodic but not strongly aperiodic SFT on $BS(1,n)$.*

Proof. First, in the $n \geq 3$ case, there is a weakly periodic configuration in X^{τ_σ}, see Lemma 8.26. Hence it is not a strongly aperiodic SFT.

In the $n = 2$ case, we define u and v the two fixpoints of ${\sigma_1}^2$ (Lemma 8.23 again) and remark that $v = \sigma_1(u)$ and $u = \sigma_1(v)$. We define a configuration $x \in {\tau_\sigma}^{BS(1,n)}$ by:

$$x_{b^{-k}a^l} := \begin{cases} (u_l, 1) \text{ if } k+m \equiv 0 \bmod 2 \\ (v_l, 1) \text{ if } k+m \equiv 1 \bmod 2 \end{cases}$$

$$x_{b^{-k}a^lb^m} := \begin{cases} (u_{F^m(l)}, R\circ F^{m-1}(l)) \text{ for } m > 0 \text{ if } k+m \equiv 0 \bmod 2 \\ (v_{F^m(l)}, R\circ F^{m-1}(l)) \text{ for } m > 0 \text{ if } k+m \equiv 1 \bmod 2 \end{cases}$$

and we use the same notations as in the proof of Lemma 8.26. The reasoning is also the same, except instead of using w an alternation appears between u and v in all the equations. As a consequence, the configuration is b^2-periodic instead of b. Once again, X^{τ_σ} is consequently not strongly aperiodic.

Now, using Proposition 8.25, and since all powers of a are of infinite order in $BS(1,n)$, we get that for any valid configuration x of τ_σ, $|Orb_{BS(1,n)}(x)| = +\infty$, for any $n \geq 2$. Hence no configuration of τ_σ is strongly periodic, and so the SFT is weakly aperiodic. □

8.3 A strongly aperiodic SFT on $BS(n,n)$

This section is a mere assembly of known results, that we nonetheless think are worth gathering in the context of this part of the present thesis. Indeed, they lead to Theorem 20 – a somewhat straightforward result that may be considered folklore by specialists, but whose underlying reasoning has not been properly written down elsewhere[1].

We begin by using a theorem from [Jea15] seen as an extension of the construction presented in [Kar07]. The idea behind that theorem is that $G \times \mathbb{Z}$ admits a strongly aperiodic SFT as soon as G can encode piecewise affine functions. This is reflected by the PA' condition described in [Jea15] and restated below.

Definition 8.27. Let $k \in \mathbb{N}$. Let $\mathcal{F} = \{f_i \colon P_i \to P'_i \mid i \in \{0, \ldots, k\}\}$ be a finite set of piecewise affine rational homeomorphisms, where each P_i and P'_i is a finite union of bounded rational polytopes of $\mathbb{R}^n$. Let $D = \bigcap_{i=1}^{k} P_i \cap \bigcap_{i=1}^{k} P'_i$ be the common domain of all functions of $\mathcal{F}$ and their inverses.

Let $S_{\mathcal{F}}$ be the closure of the set $\{f_i, {f_i}^{-1} \mid i \in \{1, \ldots, k\}\}$ under composition. We define $G_{\mathcal{F}}$, the group $\{f|_D \mid f \in S_{\mathcal{F}}\}$.

A finitely generated group G is PA'-recognizable if there exists a finite set $\mathcal{F}$ of piecewise affine rational homeomorphisms such that:

(A) $G \cong G_{\mathcal{F}}$;

(B) $\forall t \in D, \forall g \in \mathcal{F}, [\forall f \in \mathcal{F}, g(f(t)) = f(t)] \Rightarrow g = Id$.

Theorem 19 ([Jea15], Th. 7)**.** *If G is an infinite finitely generated PA'-recognizable group, then $\mathbb{Z} \times G$ admits a strongly aperiodic SFT.*

We need two additional propositions to obtain the desired result on $BS(n, n)$:

Proposition 8.28 ([CP15], Prop. 9 & 10)**.** *If G is a finitely generated group and H is a finitely generated subgroup of G of finite index, then we have the following:*

$$H \text{ admits a weakly aperiodic SFT} \Leftrightarrow G \text{ admits a weakly aperiodic SFT}$$
$$H \text{ admits a strongly aperiodic SFT} \Leftrightarrow G \text{ admits a strongly aperiodic SFT.}$$

The following proposition is known, but we include a self-contained proof.

Proposition 8.29. *$BS(n, n)$ admits $\mathbb{Z} \times \mathbb{F}_n$ as a subgroup of finite index, where $\mathbb{F}_n$ is the free group of order n.*

Proof. Let H be the subgroup of $BS(n, n)$ generated by $\{a^n\} \cup \{a^i b a^{-i} \mid i \in \{0, \ldots, n-1\}\}$. First, H is normal in $BS(n, n)$. We prove that $aHa^{-1} \subseteq H$ by verifying it on its generators: the only verification needed is $aa^{n-1}ba^{-(n-1)}a^{-1} = a^n b a^{-n} = ba^n a^{-n} = b$. Similarly, $a^{-1}Ha \subseteq H$; and finally, $bHb^{-1} \subseteq H$ (same for b^{-1}) since $b \in H$. Second, H is isomorphic to $\mathbb{Z} \times \mathbb{F}_n$ through the following isomorphism (denoting $g_0, \ldots, g_{n-1}$ the generators of $\mathbb{F}_n$ and ε its identity):

$$\begin{aligned} \varphi \colon \mathbb{Z} \times \mathbb{F}_n &\longrightarrow H \\ (1, \varepsilon) &\longmapsto a^n \\ (0, g_i) &\longmapsto a^i b a^{-i} \end{aligned}$$

It is a morphism by construction, which is correctly defined since the only basic relation of $\mathbb{Z} \times \mathbb{F}_n$, that is $(1, \varepsilon) \cdot (0, g_i) = (0, g_i) \cdot (1, \varepsilon)$, is preserved in H: $a^n a^i b a^{-i} = a^i a^n b a^{-i} = a^i b a^n a^{-i} = a^i b a^{-i} a^n$.

[1]That being said, the present result is strictly included in a result of strong aperiocity on Generalized Baumslag-Solitar groups, in an article [ABHT22] by Aubrun, Bitar and Huriot-Tattegrain, recent at the time of writing.

Said morphism is surjective, because H is generated by a^n and $\{a^iba^{-i} \mid i \in \{0,\dots,n-1\}\}$. Finally, it is also injective: let $g = (k,w) \in \mathbb{Z} \times \mathbb{F}_n$, with $w = (g_{i_1})^{e_1} \dots (g_{i_N})^{e_N}$ where the e_l are in $\{-1,+1\}$.

$$\varphi(g) = e \Leftrightarrow a^{nk}a^{i_1}b^{e_1}a^{-i_1}a^{i_2} \dots a^{-i_{N-1}}a^{i_N}b^{e_N}a^{-i_N} = e$$

This form is a canonical form in H: any word in H can be uniquely written as such. Indeed, any word in H is a succession of generators of it, $a^{i_k}b^{e_k}a^{-i_k}$ and a^n. But a^n commutes with all the other generators due to the relation of $BS(n,n)$, so such a form is always attainable. To prove it is unique, it is enough to prove it for e: suppose we have some $a^{nk}a^{i_1}b^{e_1}a^{-i_1}a^{i_2} \dots a^{-i_{N-1}}a^{i_N}b^{e_N}a^{-i_N} = e$. First, realize that no relation in $BS(n,n)$ allows to reduce the total power of a in a word, causing $k = 0$ necessarily. Then, consider the resulting word $a^{i_1}b^{e_1}a^{i_2-i_1} \dots a^{i_N-i_{N-1}}b^{e_N}a^{-i_N}$ in $BS(n,n)$: it cannot be reduced in $BS(n,n)$ since all powers of a between two b's are of absolute value smaller than n.

As a consequence, the previous equality is true only when $k = 0$ and $w = \varepsilon$. Hence the injectivity of the map.

Moreover, any element of $BS(n,n)$ can be written in a form that much resembles the one mentioned above:

$$a^pa^{nk}a^{i_1}b^{e_1}a^{-i_1} \dots a^{i_N}b^{e_N}a^{-i_N}$$

with $p \in \{0,\dots,n-1\}$. To do so, first move all a^n's in the rightmost power of a in the word, to the leftmost part of the word. Ensure that $-i_N$, the remaining power, is in $\{-n+1,\dots,-1,0\}$. Then force a^{i_N} to appear on the left of the b itself to the left of a^{-i_N}, and call $-i_{N-1}$ the remaining power of a (it is in $\{-(n-1),\dots,-1,0\}$ up to moving another a^n to the leftmost part of the word) before another b to the left. Repeat this operation until there is no b to the left of the power of a you consider, and split this final a^K into $a^pa^{nk}a^{i_1}$.

As a consequence, $BS(n,n)/H = \{\overline{e}, \overline{a}, \dots, \overline{a^{n-1}}\} \cong \mathbb{Z}/n\mathbb{Z}$. Hence H is of finite index in $BS(n,n)$. ☐

Theorem 20. *For every $n \geq 2$, $BS(n,n)$ admits a strongly aperiodic SFT.*

Proof. First, finitely generated subgroups of compact groups of matrices on rationals are PA'-recognizable (see [Jea15], Proposition 5.12). $\mathbb{F}_2$, the free group of order 2, is isomorphic[2] to a subgroup of $SO_3(\mathbb{Q})$, hence it is PA'-recognizable. It is also known (see [CSC10, Corollary D.5.3]) that $\mathbb{F}_n$ is a subgroup of $\mathbb{F}_2$; so it is isomorphic to a subgroup of $SO_3(\mathbb{Q})$ and PA'-recognizable too. Therefore by Theorem 19 $\mathbb{Z} \times \mathbb{F}_n$ admits a strongly aperiodic SFT. Using Proposition 8.28 and Proposition 8.29, we obtain that $BS(n,n)$ admits a strongly aperiodic SFT. ☐

[2]The proof will not be given here for concision; it is a key element in the proof of the Banach-Tarski paradox. The generators most often considered are $\begin{pmatrix} 3/5 & 4/5 & 0 \\ -4/5 & 3/5 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 3/5 & 4/5 \\ 0 & -4/5 & 3/5 \end{pmatrix}$.

Bibliography

[ABHT22] Nathalie Aubrun, Nicolás Bitar, and Sacha Huriot-Tattegrain. Strongly aperiodic sfts on generalized baumslag-solitar groups. In *arXiv preprint arXiv:2204.11492*, 2022.

[ABJ18] Nathalie Aubrun, Sebastián Barbieri, and Emmanuel Jeandel. About the domino problem for subshifts on groups. In *Sequences, Groups, and Number Theory*, pages 331–389. Springer International Publishing, 2018.

[ABM19] Nathalie Aubrun, Sebastián Barbieri, and Etienne Moutot. The domino problem is undecidable on surface groups. In *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany.*, pages 46:1–46:14, 2019.

[AK13] Nathalie Aubrun and Jarkko Kari. Tiling Problems on Baumslag-Solitar groups. In *MCU'13*, pages 35–46, 2013.

[AK21] Nathalie Aubrun and Jarkko Kari. Addendum to "tiling problem on baumslag-solitar groups". In *arXiv preprint arXiv:1506.06492*, 2021.

[Aki93] Ethan Akin. *The general topology of dynamical systems*, volume 1 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 1993.

[AS13] Nathalie Aubrun and Mathieu Sablik. Simulation of effective subshifts by two-dimensional subshifts of finite type. In *Acta applicandae mathematicae*, volume 126, pages 35–63. Springer, 2013.

[Bar19] Sebastián Barbieri. On the entropies of subshifts of finite type on countable amenable groups. In *arXiv preprint arXiv:1905.10015*, 2019.

[Bar20] Laurent Bartholdi. Monadic second-order logic and the domino problem on self-similar graphs. In *arXiv preprint arXiv:2011.02735*, 2020.

[BDP$^+$15] Laurent Boyer, Martin Delacourt, Victor Poupet, Mathieu Sablik, and Guillaume Theyssier. μ-limit sets of cellular automata from a computational complexity perspective. In *Journal of Computer and System Sciences*, volume 81, pages 1623–1647. Elsevier, 2015.

[BDS10] Laurent Boyer, Martin Delacourt, and Mathieu Sablik. Construction of μ-limit sets. In Jarkko Kari, editor, *Second Symposium on Cellular Automata "Journées Automates Cellulaires", JAC 2010, Turku, Finland, December 15-17, 2010. Proceedings*, pages 76–87. Turku Center for Computer Science, 2010.

[Ber66] Robert Berger. *Undecidability of the Domino Problem*, volume 66 of *Memoirs AMS*. American Mathematical Society, 1966.

[BGK11] Alexis Ballier, Pierre Guillon, and Jarkko Kari. Limit sets of stable and unstable cellular automata. In *Fundamenta Informaticae*, volume 110, pages 45–57. IOS Press, 2011.

[BMP18] Raimundo Briceño, Kevin McGoff, and Ronnie Pavlov. Factoring onto $\mathbb{Z}^d$ subshifts with the finite extension property. In *Proceedings of the American Mathematical Society*, volume 146, pages 5129–5140, 2018.

[BMP22] Robert Bland, Kevin McGoff, and Ronnie Pavlov. Subsystem entropies of shifts of finite type and sofic shifts on countable amenable groups. In *arXiv preprint arXiv:2201.01991*, 2022.

[BP09] Marie-Pierre Béal and Dominique Perrin. Completing codes in a sofic shift. In *Theoretical Computer Science*, volume 410, pages 4423–4431, 2009.

[BPS10] Mike Boyle, Ronnie Pavlov, and Michael Schraudner. Multidimensional sofic shifts without separation and their factors. In *Transactions of the American Mathematical Society*, volume 362, pages 4617–4653, 2010.

[BS16] Sebastián Barbieri and Mathieu Sablik. The domino problem for self-similar structures. In Arnold Beckmann, Laurent Bienvenu, and Nataša Jonoska, editors, *Pursuit of the Universal*, pages 205–214, Cham, 2016. Springer International Publishing.

[BS18] Alexis Ballier and Maya Stein. The domino problem on groups of polynomial growth. In *Groups, Geometry, and Dynamics*, volume 12, pages 93–105, 2018.

[BT00] François Blanchard and Pierre Tisseur. Some properties of cellular automata with equicontinuity points. In *Annales de l'Institut Henri Poincaré (B) Probabilités et Statistiques*, volume 36, pages 569–582. Institute Henri Poincaré, 2000.

[Cas00] Julien Cassaigne. Subword complexity and periodicity in two or more dimensions. In *Developments In Language Theory: Foundations, Applications, and Perspectives*, pages 14–21. World Scientific, 2000.

[CGS17] David Bruce Cohen and Chaim Goodman-Strauss. Strongly aperiodic subshifts on surface groups. In *Groups, Geometry, and Dynamics*, volume 11, pages 1041–1059, 2017.

[CI96] Karel Culik II. An aperiodic set of 13 wang tiles. In *Discrete Mathematics*, volume 160, pages 245–251. Elsevier, 1996.

[CIPY89] Karel Culik II, Jan Pachl, and Sheng Yu. On the limit sets of cellular automata. In *SIAM Journal on Computing*, volume 18, pages 831–842. SIAM, 1989.

[CN06] Ethan Coven and Zbigniew Nitecki. On the genesis of symbolic dynamics as we know it. In *Colloquium Mathematicum*, volume 110, 12 2006.

[CN10] J. Cassaigne and F. Nicolas. Factor complexity. In Valérie Berthé and Michel Rigo, editors, *Combinatorics, Automata and number Theory*, Encyclopedia of Mathematics and its Applications, page 163–247. Cambridge University Press, 2010.

[Coh17] David Bruce Cohen. The large scale geometry of strongly aperiodic subshifts of finite type. In *Advances in Mathematics*, volume 308, pages 599–626. Elsevier, 2017.

[CP15] David Carroll and Andrew Penland. Periodic points on shifts of finite type and commensurability invariants of groups. In *arXiv preprint arXiv:1502.03195*, 2015.

[CSC10] Tullio Ceccherini-Silberstein and Michel Coornaert. Cellular automata. In *Cellular Automata and Groups*, pages 1–36. Springer, 2010.

[Del21] Martin Delacourt. Rice's theorem for generic limit sets of cellular automata. In Alonso Castillo-Ramirez, Pierre Guillon, and Kévin Perrot, editors, *27th IFIP WG 1.5 International Workshop on Cellular Automata and Discrete Complex Systems (AUTOMATA 2021)*, volume 90 of *Open Access Series in Informatics (OASIcs)*, pages 6:1–6:12, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[Des06] Angela Desai. Subsystem entropy for $\mathbb{Z}^d$ sofic shifts. In *Indagationes Mathematicae*, volume 17, pages 353–359. Elsevier, 2006.

[DG19] Saliha Djenaoui and Pierre Guillon. The generic limit set of cellular automata. In *Journal of Cellular Automata*, volume 14, pages 435–477, 2019.

[DGG14] Bruno Durand, Guilhem Gamard, and Anaël Grandjean. Aperiodic tilings and entropy. In *International Conference on Developments in Language Theory*, pages 166–177. Springer, 2014.

[DMS18] Benjamin Hellouin De Menibus and Mathieu Sablik. Characterization of sets of limit measures of a cellular automaton iterated on a random configuration. In *Ergodic Theory and Dynamical Systems*, volume 38, pages 601–650. Cambridge University Press, 2018.

[DPST11] Martin Delacourt, Victor Poupet, Mathieu Sablik, and Guillaume Theyssier. Directional dynamics along arbitrary curves in cellular automata. In *Theoretical Computer Science*, volume 412, pages 3800–3821. Elsevier, 2011.

[DRS12] Bruno Durand, Andrei Romashchenko, and Alexander Shen. Fixed-point tile sets and their applications. In *Journal of Computer and System Sciences*, volume 78, pages 731–764. Elsevier, 2012.

[FW65] N. J. Fine and H. S. Wilf. Uniqueness theorems for periodic functions. In *Proceedings of the American Mathematical Society*, volume 16, pages 109–114, 1965.

[GdM19] Silvère Gangloff and Benjamin Hellouin de Menibus. Effect of quantified irreducibility on the computability of subshift entropy. In *Discrete and Continuous Dynamical Systems*, volume 39, pages 1975–2000. American Institute of Mathematical Sciences (AIMS), Springfield, MO, 2019.

[GR10] Pierre Guillon and Gaétan Richard. Asymptotic behavior of dynamical systems and cellular automata. In *arXiv preprint arXiv:1004.4743*, 2010.

[GS10] Chaim Goodman-Strauss. A hierarchical strongly aperiodic set of tiles in the hyperbolic plane. In *Theoretical Computer Science*, volume 411, pages 1085–1093. Elsevier, 2010.

[GS18] Silvère Gangloff and Mathieu Sablik. Simulation of minimal effective dynamical systems on the cantor sets by minimal tridimensional subshifts of finite type. In *arXiv preprint arXiv:1806.07799*, 2018.

[GS21] Silvère Gangloff and Mathieu Sablik. Quantified block gluing for multidimensional subshifts of finite type: aperiodicity and entropy. In *Journal d'Analyse Mathématique*, volume 144, pages 21–118. Springer, 2021.

[HdMS18] Benjamin Hellouin de Menibus and Mathieu Sablik. Characterization of sets of limit measures of a cellular automaton iterated on a random configuration. In *Ergodic Theory and Dynamical Systems*, volume 38, pages 601–650, 2018.

[HM10] Michael Hochman and Tom Meyerovitch. A characterization of the entropies of multidimensional shifts of finite type. In *Annals of Mathematics*, volume 171, pages 2011–2038. JSTOR, 2010.

[Hoc16] Michael Hochman. Multidimensional shifts of finite type and sofic shifts. In Valérie Berthé and Michel Rigo, editors, *Combinatorics, Words and Symbolic Dynamics*, Encyclopedia of Mathematics and its Applications, page 296–358. Cambridge University Press, 2016.

[Hur87] Lyman P. Hurd. Formal language characterizations of cellular automaton limit sets. In *Complex Systems*, volume 1, pages 69–80, 1987.

[Hur90a] Mike Hurley. Attractors in cellular automata. In *Ergodic Theory and Dynamical Systems*, volume 10, pages 131–140. Cambridge University Press, 1990.

[Hur90b] Mike Hurley. Ergodic aspects of cellular automata. In *Ergodic Theory and Dynamical Systems*, volume 10, pages 671–685. Cambridge University Press, 1990.

[Hur92] Mike Hurley. Attractors in restricted cellular automata. In *Proceedings of the American Mathematical Society*, volume 115, pages 563–571. American Mathematical Society, 1992.

[Jea15] Emmanuel Jeandel. Aperiodic Subshifts of Finite Type on Groups. In *arXiv preprint arXiv:1501.06831*, 2015.

[JKM07] Aimee Johnson, Steve Kass, and Kathleen Madden. Projectional entropy in higher dimensional shifts of finite type. In *Complex Systems*, volume 17, pages 243–257, 2007.

[JR15] Emmanuel Jeandel and Michael Rao. An aperiodic set of 11 wang tiles. In *arXiv preprint arXiv:1506.06492*, 2015.

[Kar92] Jarkko Kari. The nilpotency problem of one-dimensional cellular automata. In *SIAM Journal on Computing*, volume 21, pages 571–586. SIAM, 1992.

[Kar94] Jarkko Kari. Rice's theorem for the limit sets of cellular automata. In *Theoretical Computer Science*, volume 127, pages 229 − 254, 1994.

[Kar96] Jarkko Kari. A small aperiodic set of wang tiles. In *Discrete Mathematics*, volume 160, pages 259–264, 1996.

[Kar07] Jarkko Kari. The tiling problem revisited. In *International Conference on Machines, Computations, and Universality*, pages 72–79. Springer, 2007.

[Kec95] Alexander S. Kechris. *Classical descriptive set theory*, volume 156 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995.

[Ken02] R. Kenyon. An Introduction to the Dimer Model. In *arXiv preprint math/0310326*, pages 271–274, 2002.

[KM00] Petr Kůrka and Alejandro Maass. Limit sets of cellular automata associated to probability measures. In *Journal of Statistical Physics*, volume 100, pages 1031–1047, 2000.

[KM19] Jarkko Kari and Etienne Moutot. Nivat's conjecture and pattern complexity in algebraic subshifts. In *Theor. Comput. Sci.*, volume 777, pages 379–386, 2019.

[KM21] Jarkko Kari and Etienne Moutot. Decidability and periodicity of low complexity tilings. In *Theory of Computing Systems*, pages 1–24. Springer, 2021.

[Kůr97] Petr Kůrka. Languages, equicontinuity and attractors in cellular automata. In *Ergodic Theory and Dynamical Systems*, volume 17, pages 417–433. Cambridge University Press, 1997.

[Kůr03] Petr Kůrka. *Topological and symbolic dynamics*. Société mathématique de France Paris, 2003.

[Lin89] Douglas Lind. Perturbations of shifts of finite type. In *SIAM journal on discrete mathematics*, volume 2, pages 350–365. SIAM, 1989.

[LM95] Douglas Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, New York, NY, USA, 1995.

[LMP13] Erez Louidor, Brian Marcus, and Ronnie Pavlov. Independence entropy of $\mathbb{Z}^d$-shift spaces. In *Acta Applicandae Mathematicae*, volume 126, pages 297–317. Springer, 2013.

[Lot97] M. Lothaire. *Combinatorics on words*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, 1997.

[LS01] Roger C. Lyndon and Paul E. Schupp. *Combinatorial group theory*. Classics in mathematics. Springer, 2001.

[Maa95] Alejandro Maass. On the sofic limit sets of cellular automata. In *Ergodic Theory and Dynamical Systems*, volume 15, pages 663–684. Cambridge University Press, 1995.

[Mar08] Maurice Margenstern. The domino problem of the hyperbolic plane is undecidable. In *Theoretical Computer Science*, volume 407, pages 29–84. Elsevier, 2008.

[Mes72] Stephen Meskin. Nonresidually finite one-relator groups. In *Transactions of the American Mathematical Society*, volume 164, pages 105–114, 1972.

[MH38] Marston Morse and Gustav A Hedlund. Symbolic dynamics. In *American Journal of Mathematics*, volume 60, pages 815–866. JSTOR, 1938.

[Mil85] John Milnor. On the concept of attractor. In *Communications in Mathematical Physics*, volume 99, pages 177–195, 06 1985.

[Moz89] Shahar Mozes. Tilings, substitution systems and dynamical systems generated by them. In *Journal d'analyse mathématique*, volume 53, pages 139–186. Springer, 1989.

[Pan86] Jean-Jacques Pansiot. Decidability of periodicity for infinite words. In *RAIRO-Theoretical Informatics and Applications*, volume 20, pages 43–46. EDP Sciences, 1986.

[Pia08] Steven T. Piantadosi. Symbolic dynamics on free groups. In *Discrete and Continuous Dynamical Systems - A*, volume 20, pages 725–738, 2008.

[PS15] Ronnie Pavlov and Michael Schraudner. Entropies realizable by block gluing $\mathbb{Z}^d$ shifts of finite type. In *Journal d'Analyse Mathématique*, volume 126, pages 113–174. Springer, 2015.

[Rob71] Raphael M Robinson. Undecidability and nonperiodicity for tilings of the plane. In *Inventiones mathematicae*, volume 12, pages 177–209. Springer-Verlag, 1971.

[SSU20] Ayse A. Sahin, Michael Schraudner, and Ilie Ugarcovici. A strongly aperiodic shift of finite type on the discrete heisenberg group using robinson tilings. In *arXiv preprint arXiv:2009.07751*, 2020.

[SW02] Jeffrey Shallit and Ming-wei Wang. On two-sided infinite fixed points of morphisms. In *Theoretical Computer Science*, volume 270, pages 659–675. Elsevier, 2002.

[Taa07] Siamak Taati. Cellular automata reversible over limit set. In *Journal of Cellular Automata*, volume 2, pages 167–177, 2007.

[Tör20] Ilkka Törmä. Complexity of generic limit sets of cellular automata. In Hector Zenil, editor, *Cellular Automata and Discrete Complex Systems (AUTOMATA 2020)*, volume 12286 of *Lecture Notes in Computer Science*, pages 126–138. Springer International Publishing, 2020.

[Tör21] Ilkka Törmä. Generically nilpotent cellular automata. In Anni Hakanen, Vesa Halava, Pyry Herva, Jarkko Kari, Tero Laihonen, Ion Petre, and Aleksi Saarela, editors, *Proceedings of the Sixth Russian-Finnish Symposium on Discrete Mathematics (RuFiDiM 2021)*, volume 31 of *TUCS Lecture Notes*, pages 142–156. Turku Centre for Computer Science, 2021.

[Wan61] Hao Wang. Proving theorems by pattern recognition—ii. In *Bell system technical journal*, volume 40, pages 1–41. Wiley Online Library, 1961.

Personal Bibliography

[AES20] Nathalie Aubrun, J. Esnay, and Mathieu Sablik. Domino Problem Under Horizontal Constraints. In Christophe Paul and Markus Bläser, editors, *37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020)*, volume 154 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:15, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[EM22] Solène J. Esnay and Etienne Moutot. Aperiodic SFTs on Baumslag-Solitar groups. In *Theoretical Computer Science*, volume 917, pages 31–50, 2022.

[ENT22] Solène J. Esnay, Alonso Núñez, and Ilkka Törmä. Arithmetical complexity of the language of generic limit sets of cellular automata. In *arXiv preprint arXiv:2204.06215*, 2022.

[ES22] Solène J. Esnay and Mathieu Sablik. Parametrization by horizontal constraints in the study of algorithmic properties of $\mathbb{Z}^2$-subshift of finite type. In *arXiv preprint arXiv:2203.00434*, 2022.