



HAL
open science

Graph Subshifts

Amélia Durbec

► **To cite this version:**

Amélia Durbec. Graph Subshifts. 27th International Workshop on Cellular Automata and Discrete Complex Systems, Jul 2021, Marseille, France. hal-03238087v2

HAL Id: hal-03238087

<https://hal.science/hal-03238087v2>

Submitted on 19 Jun 2021 (v2), last revised 17 Feb 2023 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Graph Subshifts

Amélia Durbec¹

¹Aix-Marseille Université, Université de Toulon, CNRS, LIS, Marseille, France

May 5, 2021

Abstract

We propose a definition of graph subshifts of finite type that extends the power of both subshifts of finite type from classical symbolic dynamics and finitely presented groups from combinatorial group theory. These are sets of (finite or infinite) graphs that are defined by forbidding finitely many *local patterns*. The definition makes it not obvious to forbid finite graphs, but we prove that some nontrivial subshifts actually contain only infinite graphs: these are built either by forcing aperiodicity (such as in classical constructions of subshifts of finite type on the grid), or no residual finiteness of the period group.

1 Introduction

Subshift of finite types are well studied objects in symbolic dynamics [8] and ergodic theory. Given an alphabet Σ , a subshift of finite type (SFT) on \mathbb{Z}^d is a set of configurations $\mathcal{Y} \subseteq \Sigma^{\mathbb{Z}^d}$ that do not contain a finite set of forbidden patterns $\mathcal{F} \subseteq \Sigma^{\{-n, n\}^d}$. Despite their relatively benign and local definition, SFT have proven to have complex global behaviors as natural problems such as determining if the SFT contains a periodic configuration are proven to be undecidable [4]. This is a characteristic that is shared by multiple dynamical systems and notably with cellular automata [7]. However, the two models are deeply connected as cellular automata of dimension d can be seen as subshift of dimension $d + 1$. This connection has been used to prove multiple theorems on different subclasses of cellular automata [6]. This result has been one of the original motivators for this paper, as Graph Subshifts can be seen as a means to study a generalization of cellular automata : causal graph dynamics (CGD) [1]. CGD extends cellular automata in two complementary ways, they are defined on all graphs with degree below a given integer m , but they also allow the graph itself to evolve according to a local, shift-invariant rule. Subshifts have already been generalized to subshifts on finitely presented groups, however this is not enough to simulate CGD as all configurations share the same graph as canvas. In this paper, we provide a formalism relaxing this constraint, allowing the support to be directly dependent on the set of forbidden patterns. Furthermore, we study the presence of finite configurations and show that their presence is closely tied to different areas of mathematics : periodicity from symbolic dynamics, residually finite groups from combinatorial group theory [3][9], and graph covers which are widely used in graph theory and distributed computing.

2 Graphs

Graphs. Our graphs are the usual, connected, undirected, possibly infinite, bounded-degree labeled graphs, but with a few added twists:

- Vertices are connected through their ports: an edge is an unordered pair $\{u : a, v : b\}$, where u, v are vertices and $a, b \in \pi$ are ports. Each port is used at most once per

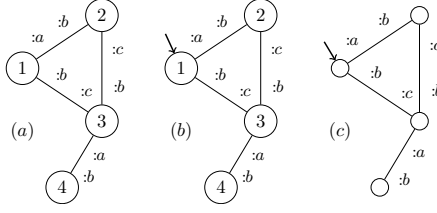


Figure 1: *The different types of graphs.* (a) A graph G . (b) A pointed graph $(G, 1)$. (c) A pointed graph modulo X .

node: if both $\{u : a, v : b\}$ and $\{u : a, v' : b'\}$ are edges, then $v = v'$ and $b = b'$. As a consequence the degree of the graph is bounded by $|\pi|$.

- A vertex u is given a label $\sigma(u)$, taken within a finite set Σ , also referred to as an internal state.

The set of all graphs having ports π , vertex labels Σ is denoted $\mathcal{G}_{\Sigma, \pi}$ (β will also be fixed within such a set, implicitly) or simply \mathcal{G} .

Pointed graphs modulo. There is actually one more added twist, so that the set of graphs becomes a compact space. Indeed, consider a graph G , a vertex $v \in V(G)$, and $D = G_v^r$ the disk of radius r around v , i.e. the subgraph of G induced by the r -neighbors of v in G . Now forgetting about G , this D is one amongst the many possible disks of radius r that can appear in \mathcal{G} . In order for the set of graphs to form a compact space, one requirement is that the set of disks of radius r be finite. Morally this is the case, since the degree is bounded by $|\pi|$. Vertex names jeopardize this however; we must get rid of them. Thus,

- Graphs are endowed with a privileged pointed vertex playing the role of an origin.
- These pointed graphs are considered modulo isomorphism, so that only the relative positions of vertices matter.

The set of all pointed graphs modulo is denoted $\mathcal{X}_{\Sigma, \pi}$ or simply \mathcal{X} . Given a graph non-modulo G , and a vertex $v \in V(G)$, we write $X((G, v))$ for the corresponding pointed graph modulo.

Paths and vertices. Over graphs modulo isomorphism without pointer, vertices no longer have a unique identifier, which makes designating a vertex a daunting task. Fortunately, our graphs are pointed, so that any vertex of the graph can be designated by the sequence of ports that leads from the origin to this vertex, i.e. elements of Π^* with $\Pi = \pi^2$. For instance, if u designates a vertex in $V(X)$ and if there is an edge $\{u : a, v : b\} \in E(G)$, then $u.ab$ designates vertex $v \in V(X)$. If there is no such edge, $u.ab$ designates an ‘undefined’ vertex \perp . Notice that, starting from origin, following path $v = (ab) \cdots (cd)$ and then path $\bar{v} = (dc) \cdots (ba)$, leads back to the origin ε .

Shifts and disks. Given a pointed graph modulo X , X_u designates the shifted version of X , where the pointer has been moved along path u . Given a pointed graph modulo X , X^r denotes the disk of radius r around the pointer. X_u^r reads $(X_u)^r$, it is the subdisk of radius r around the vertex designated by the path u . The set of all disks of radius r is denoted $\mathcal{X}_{\Sigma, \pi}^r$ or simply \mathcal{X}^r .

Operations on graphs non-modulo. Sometimes we still need to manipulate usual graphs, where vertices do have names, typically in order to perform unions of two graphs in a way that specifies their overlap. For this purpose we use a canonical naming function $X \in \mathcal{X} \mapsto G(X) \in \mathcal{G}$, which names each vertex of X by the set of paths that lead to it, starting from ε . This $G(X)$ is referred to as ‘the associated graph’. The union of two graphs G, G' is obtained by identification of the vertices $v \in G, v' \in G'$ whose names are equal. For

this union to be defined, G and G' must not disagree on identified vertices, i.e. we must have $\sigma_G(v) = \sigma_{G'}(v')$ and $v.ab = v'.ab$ whenever these are defined in both the graphs. We will also need an operation $u.G$ that prefixes all the vertex names in G , with u .

Graphs, pointed graphs modulo and their operations are intuitive enough as summarized here, but their proper formalization can sometimes be rather lengthy and has been introduced several times before. This is why we pushed it to Appendix A.

3 Subshifts

Definition 1 (Cuts and cylinders). Consider $\mathcal{L} \subseteq \Pi^*$ a prefix-stable language, i.e. such that $u.v \in \mathcal{L}$ implies $u \in \mathcal{L}$. The \mathcal{L} -cut of a graph $X \in \mathcal{X}$ is the subgraph induced by the vertices labeled by prefixes of \mathcal{L} . It is denoted $X|_{\mathcal{L}}$. Consider $\mathcal{Z} \subseteq \mathcal{X}$, we write $\mathcal{Z}|_{\mathcal{L}}$ for $\{X|_{\mathcal{L}} \mid X \in \mathcal{Z}\}$. Consider $Z \in \mathcal{Z}|_{\mathcal{L}}$, the cylinder of Z within \mathcal{Z} is $\{X \in \mathcal{Z} \mid X|_{\mathcal{L}} = Z\}$. It is denoted $\mathcal{Z}[Z]$.

Definition 2 (Subshift). Let \mathcal{F} be a set of tuples (F, L) , with F a finite graph and L a prefix-stable language. The **subshift** forbidding \mathcal{F} is

$$\mathcal{Z} = \{X \in \mathcal{G}_{\Sigma, \pi} \mid \forall v \in X, \forall (F, L) \in \mathcal{F}, (X_v)|_L \neq F\}.$$

Remark 1. The following are equivalent.

1. \mathcal{Z} is the subshift forbidding some finite collection \mathcal{F} .
2. \mathcal{Z} is the subshift forbidding $\mathcal{F}' \times \{M\}$ for some prefix-stable language M and some finite collection \mathcal{F}' of finite graphs.
3. \mathcal{Z} is the subshift authorizing some finite collection \mathcal{A} of pairs (A, L) where A is a finite graph and L a prefix-stable language, in the following sense:

$$\mathcal{Z} = \{X \in \mathcal{G}_{\Sigma, \pi} \mid \forall v \in X, \exists (A, L) \in \mathcal{A}, (X_v)|_L = A\}.$$

4. \mathcal{Z} is the subshift authorizing $\mathcal{A}' \times \{M\}$ for some prefix-stable language M and some finite collection \mathcal{A}' of finite graphs.

Proof.

- 1 \Leftrightarrow 2 The condition 2 is at least more restrictive, as it require for all graph to be induced from the same prefix-stable language M . However, for a given subshift \mathcal{Z} obtained from forbidding \mathcal{F} , we can choose $M = \bigcap_{(F, L) \in \mathcal{F}} L$ and $\mathcal{F}' = \{F' \mid F' \in (\mathcal{X} \setminus \mathcal{Z})|_M\}$ to obtain the same subshift.
- 3 \Leftrightarrow 4 Similarly as before, 4 is at least more restrictive then 3, but for a given subshift \mathcal{Z} obtained from authorizing \mathcal{A} we can choose $M = \bigcap_{(F, L) \in \mathcal{F}} L$ and $\mathcal{A}' = \{A' \mid A' \in \mathcal{Z}|_M\}$ to obtain the same subshift.
- 2 \Leftrightarrow 4 The subshift \mathcal{Z} obtained from forbidding $\mathcal{F}' \times M$ is the same as the one obtained from authorizing $\mathcal{A}' \times M = (\mathcal{X}|_M \setminus \mathcal{F}') \times M$.

□

Definition 3. If any property in Remark 1 occurs, then we say that \mathcal{Z} is a **subshift of finite type (SFT)**, and that it is of **order** $\min_M \delta(M)$ where $\delta(M)$ is the diameter of M .

The duality above is not true for subshifts in general, for example we can think about the set of graphs authorizing exactly those finite graphs which have exactly one red vertex, which is not closed topologically.

Remark 2. In pairs (F, L) , L cannot always be assumed to be the language labeling paths in F . For example, requiring that every vertex have an edge with port $a \in \Sigma$ (or more generally, is the start of a path with a specific label) is a very natural finite-type condition, but no forbidden pattern involves an a , so that the corresponding subshift cannot be written as $\{X \in \mathcal{G}_{\Sigma, \pi} \mid \forall v \in X, \forall F \in \mathcal{F}, (X_v)|_{\mathcal{L}_F} \neq F\}$ for any finite collection \mathcal{F} . The dual is true for a definition through a finite collection of authorized patterns: think of the (very natural) SFT consisting of those graphs forbidding port $a \in \Sigma$ (or more generally, any finite collection of words on the port alphabet).

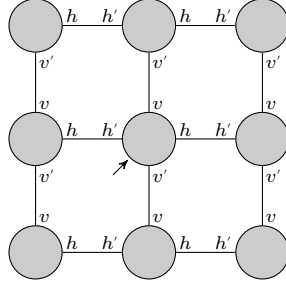


Figure 2: The locally grid-like condition can be imposed by requiring each subgraph induced to have this support on the corresponding language. The pointer is represented by a dark arrow.

4 Examples

First we are going to focus a special type of subshifts, that is **locally grid-like** subshifts. Formally this is defined by subshifts such that : the set of ports π is h, h', v, v' , all vertex have four edges, every edges are either of the form $\{u : h, v : h'\}$ or $\{u : v, v : v'\}$, and for all $u \in V(X)$ we have that $u.hh'vv' = u.vv'hh'$. This conditions are all local and easy to impose through local constraints that can be implemented in a SFT by only authorizing induced subgraph that are locally a grid on the language of a 3 by 3 grid (see Fig 2). This provides us with a first example of subshift :

Example 1 (Hard Square Model). *The hard square model subshift is define as a locally grid-like subshift $\mathcal{Z} \subseteq \mathcal{X}_{\{0,1\},h,h',v,v'}$, such that for every configuration X and edge $\{u : a, v : b\} \in E(X)$ we have that $\sigma_x(u) = 0$ or $\sigma_x(v) = 0$. This can be easily done by forbidding 4 different graph, one for each possible edges. This is a direct generalization of a well know SFT on \mathbb{Z}^2 and other subgroup. It obviously contains all the configuration that are already present on \mathbb{Z}^2 such as Fig. 3, but also new infinite configuration such as cylinder Fig. 4, and even finite configuration such as tori only containing 0.*

Locally grid-like subshifts can be derived from any subshift on \mathbb{Z}^2 . In fact, a similar method can be applied to any subshift on any group that are finitely presented. Indeed, we can implement any relation of the group as a constraint on a graph subshift of finite type.

Example 2 (Monochromatic subshift). *We can derive graph subshift from the monochromatic full shift on $\langle a, b | abab \rangle$, by imposing two constraints (one for the generator and one on for the relation on the authorized patterns as seen in the figure 5). This subshift obviously contains the Cayley graph of $\langle a, b | abab \rangle$, but also all of the Cayley graph of a quotient subgroup (excluding really small groups such as the trivial group).*

The notion of quotient can be extended to all periodic graph, which is the subject of the following section.

5 Quotient and Periodicity

In this section we will suppose that graphs do not contains symmetric arcs, i.e arcs sharing the same ports on both ends.

Definition 4 (Homomorphism). *A (graph) **homomorphism** from graph $X \in \mathcal{X}$ to graph $X' \in \mathcal{X}$ is a function φ from $V(X)$ to $V(X')$ such that for all $u \in V(X), a, b \in \pi$,*

$$\varphi(\varepsilon) = \varepsilon, \quad \sigma_X(u) = \sigma_{X'}(\varphi(u)) \quad \text{and} \quad \varphi(u.ab) = \varphi(u).ab .$$

*It is a **quotient** if it is surjective; we say that X' **covers** X , and write $X \succeq X'$. $\max \mathcal{Y}$ will denote the maximal graph of the set $\mathcal{Y} \subseteq \mathcal{X}$ for this order, when it exists, that is $\max \mathcal{Y} \in \mathcal{Y}$ and for all $X \in \mathcal{Y}, \max \mathcal{Y} \succeq X$.*

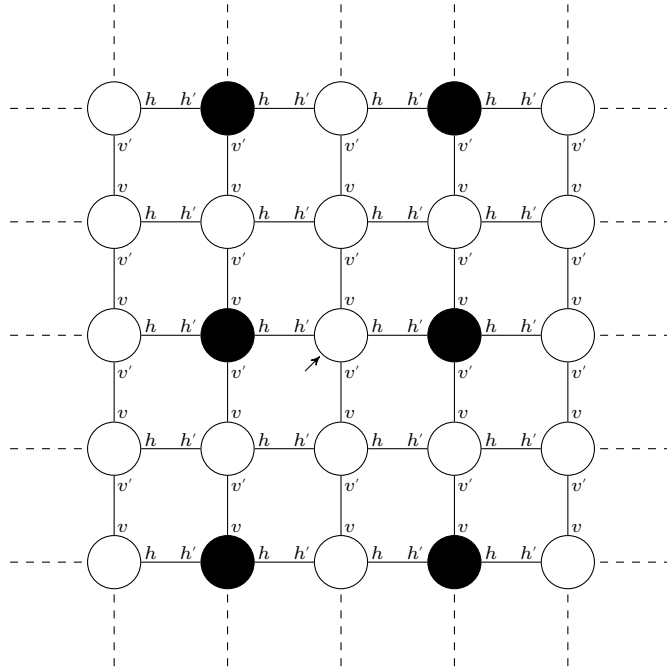


Figure 3: *Configuration of the Hardsquare model that is present in \mathbb{Z}^2 .* The color represents the states of the vertices, with the white vertices having a state 0 and the black one having the state 1.

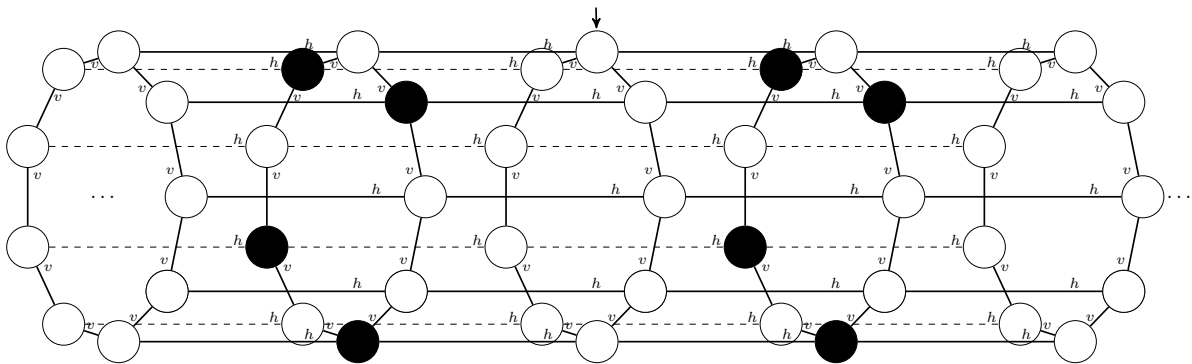


Figure 4: *Configuration of the Hardsquare model that is not present in \mathbb{Z}^2 .* The color represents the states of the vertices, with the white vertices having a state 0 and the black one having the state 1. In order to preserve clarity, only half of the ports are written and some edges have been dashed.

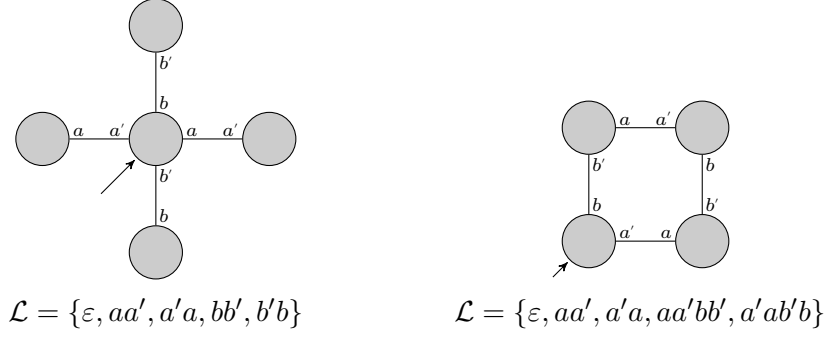


Figure 5: *Non-forbidden induced subgraphs of the monochromatic subshift extending $\langle a, b | abab \rangle$.* Each condition is verified independently, one for the generators and one for the relation. The pointer is represented by a dark arrow.

Definition 5 (Periodicity). A *period* of a graph X is a word $w \in \Pi^*$ such that $X = X_w$, it is said to be *trivial* if $w = \varepsilon$.

The *stabilizer* of X is its set of periods.

Definition 6 (Strong periodicity). A language $L \in \Pi^*$ is said to be **dense** in X if there exists $M \in \mathbb{N}$ such that for all vertex $v \in V(X)$, $d(v, \bullet L) < M$ where d is the distance in the graph.

A graph X is said to be **strongly periodic** if its stabilizer $\text{STAB}(X)$ is dense in X .

Proposition 1. For all graph X , $(\text{STAB}(X), \cdot)$ is a group, where " \cdot " denote the concatenation operation.

Proof. • Identity element : for all u , $X_{u.\varepsilon} = X_{\varepsilon.u} = X_u$

• Associativity : Let $a, b, c \in \text{STAB}(X)$. As $(X_{a.b})_c = X = (X_a)_{b.c}$

• Inverse element : Let $u \in \text{STAB}(X)$ and u^{-1} the inverse path. $X = X_{u.u^{-1}} = (X_u)_{u^{-1}} = X_{u^{-1}}$

□

Proposition 2 (Quotient graph). Let X , and H a subgroup of it's stabilizer. We can define a quotient graph $X_{/H}$ to be such that :

- $V(X_{/H}) = \{Hu | u \in X\}$ and it's pointer is at the vertex labeled H .
- $E(X_{/H}) = \{(Hu : p_1, Hv : p_2) | (u : p_1, v : p_2) \in E(X)\}$
- $\sigma_{X_{/H}}(Hu) = \sigma_X(u)$ for all $u \in X$

As for all $u \in V(X)$, uH is a subgroup of the stabilizer of X_u , this also define a quotient graph on all the graphs obtained by translating the pointer and $(X_u)_{/uH} = (X_{/H})_u$

Proof. Let us prove that $\sigma_{X_{/H}}$ is well defined. Let $u, v \in V(X)$ such that $Hu = Hv$. As H is a subgroup of the stabilizer, we have that $X_u = X_v$ and therefore $\sigma_X(u) = \sigma_X(v)$ and $\sigma_{X_{/H}}$ is well defined. Next, let us prove that $X_{/H}$ does not contains a conflict of ports. By contradiction, suppose there exist conflict of ports on the vertex $Hu \in V(X_{/H})$. First, focus on the case where the conflict is from two distinct edges $(Hu : p_1, Hv : p_2), (Hu : p_1, Hv' : p'_2) \in E(X_{/H})$ with $Hv : p_2 \neq Hv' : p'_2$. This mean there exist $(u : p_1, v : p_2) \in E(X_{/H})$ and $(u' : p_1, v' : p'_2) \in E(X_{/H})$ such that $u' \in Hu$. As $u' \in Hu$, we have that $X_u = X_{u'}$ and therefore we have that $p_1 = p_2$ and we can write $v = u.p_1p_2$ and $v' = u'.p_1p'_2$. As $u' \in Hu$ there exist $h \in H$ such that $v' = h.u.p_1p'_2 = h.v$ and $Hv = Hv'$. This contradicts the fact that $(Hu : p_1, Hv : p_2), (Hu : p_1, Hv' : p'_2)$ are distinct edges. Now, suppose that the conflict is from a self loop, i.e there exist $(Hu : p_1, Hu : p_1) \in E(X_{/H})$. Then, there exist $u' \in Hu$ such that $u.p_1p_1 = u'$. As $X_u = X_{u'}$, this means that $p_1p_1 \in H$, but this contradicts the fact that X does not contains symmetric arcs. □

Remark 3. For all graph $X \in \mathcal{X}$, and X/H quotient graph of X , X is a covering graph of X' .

Note that this definition holds even if X contains arc symmetries in the case where H do not contains axial symmetries, i.e for all $u \in H$, $u^2 \neq \varepsilon$.

Corollary 1. A graph X is strongly periodic if and only if it admits a finite quotient.

Proof. The left to right implication is consequence of the precedent remark, as $X_{/\text{STAB}(X)}$ is a finite graph. If X admits a finite quotient ϕ onto X' , then for all $u \in \phi^{-1}(\varepsilon)$ we have that $u \in \text{STAB}(X)$. $\phi^{-1}(\varepsilon)$ is dense in X as the distance of every vertex of X to a vertex of $\phi^{-1}(\varepsilon)$ is lower or equal than the radius of X' . \square

It seems likely that this theorem holds even if we allow for symmetric arcs, as strongly periodic graph have a structure too strong to have infinitely many arc symmetry, and therefore we can extract a dense subgroup \mathcal{H} such that X/H is properly defined.

Informally, another way to look at strongly periodic graph is as graphs having a factor map towards a Cayley graph, however subshift homomorphism are not as easy to describe in the classical setting.

We say that set of vertices is **r -separated**, if every pair of distinct vertices are at a distance greater than r .

Proposition 3. If every vertex of H is r -separated in X , then the quotient preserves the disks of radius r : for all $u \in V(X)$, $X_u^r = (X/H)_u^r$.

Proof. As quotients are locally bijective, we have that $(X/H)_u^r$ is at least a quotient of X_u^r . But as H is r -separated, X_u^r it contains at most one vertex of H , and therefore each vertex of $(X/H)_u^r$ has exactly one antecedent in X_u^r . \square

Theorem 1. Let \mathcal{Y} a SFT. \mathcal{Y} admits a strongly periodic configuration X such that $\text{STAB}(X)$ is residually finite if and only if it admits a finite graph.

Proof. The right to left implication is simply a consequence of the fact that finite graphs are strongly periodic and finite group are residually finite. Now, suppose that \mathcal{Y} is of order n and admits a strongly periodic configuration X such that $\text{STAB}(X)$ is residually finite. As $\text{STAB}(X)$ is residually finite, there exist a subgroup H of finite index not containing any element of X^n . As H is n -separated we can use the proposition 3 and therefore for all $u \in V(X)$ we have that $(X/H)_u^n = X_u^n$. Given that \mathcal{Y} is of order n , this means that $X/H \in \mathcal{Y}$. \square

This provides us with two complementary ways to prevent an SFT to have a finite configuration, either a SFT has no strongly periodic configuration or each of the strongly periodic configuration has a stabilizer that is not residually finite. Indeed, if the stabilizer is not residually finite then it must contains an element that is sent to ε for every possible finite quotient. But this is a local condition that can be checked and excluded from an SFT.

We can use a similar reasoning to obtain a condition on SFTs that have only one support graph (i.e all configuration X and X' , only differs by $\sigma_X \neq \sigma_{X'}$) :

Theorem 2. If an SFT \mathcal{Y} has only one support graph then for all configuration $X \in \mathcal{Y}$ we have that $\text{STAB}(X)$ is finite.

Proof. By contrapositive, suppose that X have an infinite $\text{STAB}(X)$ and note n the order of \mathcal{Y} . As $\text{STAB}(X)$ is infinite, it contains an infinite subgroup H that is r -separated. Using once again the proposition 3 therefore for all $u \in V(X)$ we have that $(X/H)_u^n = X_u^n$. Given that \mathcal{Y} is of order n , this means that $X/H \in \mathcal{Y}$. \square

This theorem will allows us to prove the following the undecidability of existence of the two following problems :

Finite configuration problem

Input: A SFT \mathcal{Z} given by a set \mathcal{F} of forbidden pattern

Output: Does \mathcal{Z} has a finite configuration?

Support graph unicity problem

Input: A SFT \mathcal{Z} given by a set \mathcal{F} of forbidden pattern

Output: Does \mathcal{Z} has only one support graph ?

Theorem 3. *The finite configuration problem is undecidable.*

Proof. Given a subshift of finite type \mathcal{Z} on \mathbb{Z}^2 given by a set of forbidden pattern, it is well know that it is undecidable to know if it aperiodic, that is if it contains an configuration that is strongly periodic. If it admit a strongly periodic configuration, then it's extension to locally grid-like admit a finite quotient. Reciprocally, if its extension to a locally grid-like subshift contains a finite configuration X then it also contains an infinite periodic configuration X' such that X is a quotient of X' . \square

Theorem 4. *The support graph unicity problem is undecidable.*

Proof. Given a subshift of finite type \mathcal{Z} on \mathbb{Z}^2 given by a set of forbidden pattern, it is undecidable to know if it is strongly aperiodic (i.e all of its configuration are not periodic). By extending \mathcal{Z} to a graph subshift of finite type \mathcal{Z}' , we have that \mathcal{Z} is strongly aperiodic if and only if \mathcal{Z}' is strongly aperiodic. If \mathcal{Z}' is strongly aperiodic then the support graph of all its configurations is unique as it is simply the infinite planar grid. Reciprocally, suppose that \mathcal{Z}' has only one support graph, then by the theorem 2 then all of its configuration have a finite stabilizer. Let X be one of this configuration, then its stabilizer must be trivial, otherwise X would be the quotient of an infinite planar grid configuration. To summarize we have that \mathcal{Z} is strongly aperiodic if and only \mathcal{Z}' has only one support graph, which concludes the proof. \square

6 Conclusion

To sum up, we provided a formalism generalizing SFT on Cayley graphs by allowing the support graph to be determined by the SFT in section 3. The generalization also allows the SFT to have finite configuration and we provided examples of such SFT. By defining quotient graphs, we have shown that the only way for an *SFT* to not have finite configurations is to only have configurations that are either aperiodic or with a stabilizer that is not residually finite. Similarly we have shown that having only one support graph require configurations to have finite stabilizers, and used both results to prove that the finite configuration problem and the support graph unicity problem are undecidable. In the future, we wish to study determinism and study directly the relation between causal graph dynamics and graph subshift, and use this connection to prove structure and complexity results on causal graph dynamics. On top of that, graph subshifts could provide a new point of view for studying group subshifts. At last, many notions and results known in the group subshifts such as sofic subshifts, determinism or entropy, would be interesting to adapt to graph subshift.

References

- [1] Pablo Arrighi, Amélia Durbec, and Aurélien Emmanuel. Reversibility vs local creation/destruction. In Michael Kirkedal Thomsen and Mathias Soeken, editors, *Reversible Computation - 11th International Conference, RC 2019, Lausanne, Switzerland, June 24-25, 2019, Proceedings*, volume 11497 of *Lecture Notes in Computer Science*, pages 51–66. Springer, 2019. doi:10.1007/978-3-030-21500-2_4.
- [2] Pablo Arrighi, Simon Martiel, and Vincent Nesme. Cellular automata over generalized cayley graphs. *Mathematical Structures in Computer Science*, 28(3):340–383, 2018. doi:10.1017/S0960129517000044.

- [3] Gilbert Baumslag and Donald Solitar. Some two-generator one-relator non-Hopfian groups. *Bulletin of the American Mathematical Society*, 68(3):199–201, 1962. 00471. URL: <https://www.ams.org/bull/1962-68-03/S0002-9904-1962-10745-9/>, doi:10.1090/S0002-9904-1962-10745-9.
- [4] Robert Berger. *The undecidability of the domino problem*. PhD, Harvard University, July 1964. 00000.
- [5] G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Math. Systems Theory*, 3:320–375, 1969.
- [6] Jarkko Kari. The nilpotency problem of one-dimensional cellular automata. *siam Journal on Computing*, 21(3):571–586, 1992.
- [7] Christopher G Langton. Self-reproduction in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):135–144, 1984.
- [8] Douglas A. Lind and Brian Marcus. *An introduction to symbolic dynamics and coding*. Cambridge University Press, Cambridge ; New York, 1995. 03085.
- [9] Roger C. Lyndon and Paul E. Schupp. *Combinatorial Group Theory*. Classics in Mathematics. Springer-Verlag, Berlin Heidelberg, 2001. 04184. URL: <https://www.springer.com/gp/book/9783540411581>.

A Formalism

This appendix provides formal definitions of the kinds of graphs we are using, together with the operations we perform upon them. It can all be found in [2] and is reproduced here only for convenience.

A.1 Graphs

Let π be a finite set, $\Pi = \pi^2$, and V some universe of names.

Definition 7 (Graph non-modulo). A graph non-modulo G is given by

- An at most countable subset $V(G)$ of V , whose elements are called vertices.
- A finite set π , whose elements are called ports.
- A set $E(G)$ of non-intersecting two element subsets of $V(G) : \pi$, whose elements are called edges. In other words an edge e is of the form $\{u : a, v : b\}$, and $\forall e, e' \in E(G), e \cap e' \neq \emptyset \Rightarrow e = e'$.
- A partial function σ from $V(G)$ to a finite set Σ .

The graph is assumed to be connected. The set of graphs with states in Σ and ports π is written $\mathcal{G}_{\Sigma, \pi}$, or simply \mathcal{G} .

We single out a vertex as the origin:

Definition 8 (Pointed graph non-modulo). A pointed graph is a pair (G, p) with $p \in V(G)$. The set of pointed graphs with states in Σ and ports π is written $\mathcal{P}_{\Sigma, \pi}$, or simply \mathcal{P} .

Here is when graph differ only up to names of vertices:

Definition 9 (Isomorphism). An isomorphism R is a function from \mathcal{G} to \mathcal{G} which is specified by a bijection $R(\cdot)$ from V to V . The image of a graph G under the isomorphism R is a graph RG whose set of vertices is $R(V(G))$, whose set of edges is $\{\{R(u) : a, R(v) : b\} \mid \{u : a, v : b\} \in E(G)\}$, and whose internal states are given by $\sigma \circ R^{-1}$. Similarly, the image of a pointed graph $P = (G, p)$ is the pointed graph $RP = (RG, R(p))$. When P and Q are isomorphic we write $P \approx Q$, defining an equivalence relation on the set of pointed graphs.

Notice that pointed graph isomorphism rename the pointer in the same way as it renames the vertex upon which it points; which effectively means that the relative position of the pointer is unchanged.

Definition 10 (Pointed graphs modulo). *Let P be a pointed (labeled) graph (G, p) . The pointed graph modulo $X(P)$ is the equivalence class of P with respect to the equivalence relation \approx . The set of labeled pointed Graphs modulo with states Σ and ports π is written $\mathcal{X}_{\Sigma, \pi}$, or simply \mathcal{X} .*

A.2 Paths and vertices

Vertices of pointed graphs modulo isomorphism can be designated by a sequence of ports in Π^* that leads, from the origin, to this vertex.

Definition 11 (Path). *Given a pointed graph modulo X , we say that $\alpha \in \Pi^*$ is a path of X if and only if there is a finite sequence $\alpha = (a_i b_i)_{i \in \{0, \dots, n-1\}}$ of ports such that, starting from the pointer, it is possible to travel in the graph according to this sequence. The set of paths of X is denoted by $L(X)$.*

Paths can be seen as words on the alphabet Π and thus come with a natural operation ‘.’ of concatenation, a unit ε denoting the empty path, and a notion of inverse path $\bar{\alpha}$ which stands for the path α read backwards. Two paths are equivalent if they designate the same vertex:

Definition 12 (Equivalence of paths). *Given a pointed graph modulo X , we define the equivalence of paths relation \equiv_X on $L(X)$ such that for all paths $\alpha, \alpha' \in V(X)$, $\alpha \equiv_X \alpha'$ if and only if, starting from the pointer, α and α' lead to the same vertex of X . We write $\hat{\alpha}$ for the equivalence class of α with respect to \equiv_X . The set of equivalence classes of paths in $L(X)$ is denoted $V(X)$.*

Given a pointed graph modulo X , it is often useful to undo the modulo, i.e. to obtain a canonical instance $(G(X), \varepsilon)$ of the equivalence class X .

Definition 13 (Associated graph). *Let X be a pointed graph modulo. Let $G(X)$ be the graph such that:*

- *The set of vertices $V(G(X))$ is $V(X)$;*
- *The edge $\{\hat{\alpha} : a, \hat{\beta} : b\}$ is in $E(G(X))$ if and only if $\alpha.ab \in L(X)$ and $\alpha.ab \equiv_X \beta$, for all $\alpha \in \hat{\alpha}$ and $\beta \in \hat{\beta}$.*
- *The internal state $\hat{\sigma}(\hat{\alpha})$ is given by $\sigma(\alpha)$.*

We say that $G(X)$ is the graph associated to X . We have that $X = X(G(X))$.

Notations. A graph modulo X and its associated graph $G(X)$ are two formalisms for the same mathematical object. A vertex this mathematical object :

- *is an equivalence class $\hat{\alpha}$ in $V(X)$, i.e. the set of all paths leading to this vertex starting from ε ,*
- *is designated by any α in $\hat{\alpha}$, i.e. any particular path leading to the vertex starting from ε .*

This led to use the following simplified mathematical conventions, throughout the paper:

- $\hat{\alpha}$ and α were not distinguished. The latter notation was used to designate the former.
- Symbols ‘ \equiv_X ’ and ‘ $=$ ’ were not distinguished. Expressions such as $\alpha = \beta$ meant equality of the vertices designated by the paths.

These simplified mathematical conventions are also used in the rest of the appendix, unless otherwise specified.

A.3 Operations over pointed Graphs modulo

Disks. For a pointed graph (G, p) non-modulo:

- the neighbors of radius r are just those vertices which can be reached in r steps starting from the pointer p ;

- the disk of radius r , written G_p^r , is the subgraph induced by the neighbors of radius r .

For a graph modulo, on the other hand, the analogous operation is:

Definition 14 (Disk). *Let $X \in \mathcal{X}$ be a pointed graph modulo and G its associated graph. Let X^r be $X(G_p^r)$. The graph modulo $X^r \in \mathcal{X}$ is referred to as the disk of radius r of X . The set of disks of radius r with states Σ and ports π is written \mathcal{X}^r .*

Shifts just move the pointer vertex:

Definition 15 (Shift). *Let $X \in \mathcal{X}$ be a pointed graph modulo and G its associated graph. Consider $u \in V(X)$ or X^r for some r , and consider the pointed graph (G, u) , which is the same as (G, ε) but with a different pointer. Let X_u be $X(G, u)$. The pointed graph modulo X_u is referred to as X shifted by u .*

A.4 Topology on pointed graphs modulo

Having a well-defined notion of disks allows us to define a topology upon \mathcal{X} , which is the natural generalization of the well-studied Cantor metric upon CA configurations [5].

Definition 16 (Gromov-Hausdorff-Cantor metrics). *Consider the function*

$$\begin{aligned} d : \mathcal{X} \times \mathcal{X} &\longrightarrow \mathbb{R}^+ \\ (X, Y) &\mapsto d(X, Y) = 0 \quad \text{if } X = Y \\ (X, Y) &\mapsto d(X, Y) = 1/2^r \quad \text{otherwise} \end{aligned}$$

where r is the minimal radius such that $X^r \neq Y^r$.

The function $d(.,.)$ is such that for $\epsilon > 0$ we have (with $r = \lfloor -\log_2(\epsilon) \rfloor$):

$$d(X, Y) < \epsilon \Leftrightarrow X^r = Y^r.$$

It defines an ultrametric distance.

Lemma 1 (Compactness). *(\mathcal{X}, d) is a compact metric space, i.e. every sequence admits a converging subsequence.*