Algorithmical and mathematical approaches of causal graph dynamics

Simon Martiel

UNIVERSITÉ NICE SOPHIA ANTIPOLIS

ECOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE

pour l'obtention du grade de

Docteur en Sciences

de l'Université Nice Sophia Antipolis

Mention: Informatique

présentée et soutenue par

Simon MARTIEL

Approches informatique et mathématique des dynamiques causales de graphes

Thèse dirigée par Pablo ARRIGHI et Bruno MARTIN

soutenue le 6 juillet 2015

Jury :

M. Pablo ARRIGHI	Université Aix-Marseille	Directeur de thèse
M. Bruno MARTIN	Université Nice Sophia Antipolis	Directeur de thèse
M. Vincent DANOS	CNRS	Rapporteur
M. Emmanuel JEANDEL	Université de Lorraine	Rapporteur
M. Jarkko KARI	University of Turku	Rapporteur
M. Gilles DOWEK	INRIA	Examinateur
M. Jean-Louis GIAVITTO	CNRS	Examinateur
M. Eric GOLES	Universidad Adolfo Ibáñez	Examinateur

Abstract

English

Cellular Automata constitute one of the most established model of discrete physical transformations that accounts for euclidean space. They implement three fundamental symmetries of physics: causality, homogeneity and finite density of information. Even though their origins lies in physics, they are widely used to model spatially distributed computation (self-replicating machines, synchronization problems,...), as well as a great variety of multi-agents phenomena (traffic jams, demographics,...). While being one of the most studied model of distributed computation, their rigidity forbids any trivial extension toward time-varying topology, which is a fundamental requirement when it comes to modelling phenomena in biology, sociology or physics: for instance when looking for a discrete formulation of general relativity. Causal graph dynamics generalize cellular automata to arbitrary, bounded degree, time-varying graphs. In this work, we generalize the fundamental structure results of cellular automata for this type of transformations. We endow our graphs with a compact metric space structure, and follow two approaches. An axiomatic approach based on the notions of continuity and shift-invariance, and a constructive approach, where a local rule is applied synchronously on every vertex of the graph. Compactness allows us to show the equivalence of these two definitions, extending the famous result of Curtis-Hedlund-Lyndons theorem. Another physics-inspired symmetry is then added to the model, namely reversibility. We answer the question whether the inverse of a causal graph dynamics is itself a causal graph dynamics, and prove the existence of a block decomposition of reversible causal graph dynamics, thereby generalizing two results of reversible cellular automata theory. We present the construction of a family of intrinsically universal local rules, indexed by the degree of the simulated rule, such that every local rule can be simulated by

one of these rules, with a constant delay, whilst preserving the space-time structure of the computation. We also present the construction of a universal construction machine, able to construct any instance of causal graph dynamics. Finally, we provide a correspondence between graphs and $\Delta-$complexes, allowing us to define causal dynamics of discrete geometrical spaces.

Français

Le modèle des automates cellulaires constitue un des modèles le mieux établi de physique discrète sur espace euclidien. Ils implantent trois symétries fondamentales de la physique: la causalité, l'homogénéité et la densité finie de l'information. Bien que l'origine des automates cellulaires provienne de la physique, leur utilisation est très répandue comme modèles de calcul distribué dans l'espace (machines auto-réplicantes, problèmes de synchronisation,...), ou bien comme modèles de systèmes multi-agents (congestion du trafic routier, études démographiques,...). Bien qu'ils soient parmis les modèles de calcul distribué les plus étudiés, la rigidité de leur structure interdit toute extension triviale vers un modèle de topologie variant dans le temps, qui se trouve être un prérequis fondamental à la modélisation de certains phénomènes biologiques, sociaux ou physiques, comme par exemple la discrétisation de la relativité générale.

Les dynamiques causales de graphes généralisent les automates cellulaires aux graphes arbitraires de degré borné et pouvant varier dans le temps. Dans cette thèse, nous nous attacherons à généraliser certains des résultats fondamentaux de la théorie des automates cellulaires. En munissant nos graphes d'une métrique compacte, nous présenterons deux approches différentes du modèle. Une première approche axiomatique basée sur les notions de continuité et d'invariance par translation, et une deuxième approche constructive, où une règle locale est appliquée en parallèle et de manière synchrone sur l'ensemble des sommets du graphe. La compacité nous permettra de prouver l'équivalence entre ces deux définitions, étendant le célèbre résultat de Curtis, Hedlund et Lyndon sur les automates cellulaires. Nous ajouterons ensuite une symmétrie supplémentaire au modèle: la réversibilité. Nous répondrons à la question de savoir si toute instance bijective de notre modèle admet bien une dynamique inverse, et nous montrerons comment toute instance réversible peut être décomposée en un circuit d'opérations locales et réversibles. Nous présenterons la construction d'une famille de règles locales intrinsèquement universelles, indexées selon le degré de la règle simulée, et capables de simuler toute autre instance du modèle, tout en préservant la structure spatio-temporelle du calcul. Ce résultat nous permettra ensuite de décrire une machine de construction universelle capable de construire n'importe quelle instance du modèle. Enfin nous étudierons une correspondance entre graphes et $\Delta-$complexes nous permettant ainsi de définir les dynamiques causales d'espaces géométriques discrets.

Remerciements

Je voudrais commencer par remercier mes deux directeurs de thèse, Pablo et Bruno. Pablo pour m'avoir fait découvrir le monde merveilleux de la recherche, la richesse des interactions entre physique et informatique et surtout pour avoir pris le temps de s'occuper d'un étudiant qui est arrivé dans son bureau, il y a maintenant six ans, en lui demandant un sujet de stage en informatique quantique. Bruno pour avoir accepté de m'encadrer sur un sujet original, pour s'être prêté au jeu des dynamiques causales de graphes et pour avoir pris le temps de relire ma thèse avec autant d'attention.

Je voudrais ensuite remercier les membres de l'équipe MC3: Enrico (pour m'avoir concocté un Master de rêve), Sandrine (si Bruno est mon "père de thèse", Sandrine est sans aucun doute ma "mère de thèse"), Christophe (pour les discussion de relativité, les jeux vidéos, et les groupes de travail sur les grands ordinaux), mon co-bureau Pierre-Alain (pour les longues discussions autout d'un thé), ainsi que ceux de passage: Émilie, Jonas et Fabien.

Je me dois de remercier mes trois rapporteurs qui ont accepté de relire cette thèse: Vincent Danos, Emmanuel Jeandel et Jarkko Kari, ainsi que mes examinateurs Jean-Louis Giavitto, Gilles Dowek et Éric Golès.

Je voudrais remercier toutes ces personnes qui ont croisé ma route au cours de ces dernières années, et avec qui j'ai eu l'occasion de collaborer ou simplement de discuter: Kevin Perrot, Simon Perdrix, Zizhu Wang, Vincent Nesme, Eric Thierry, Pascal Vanier, Maxime Senot.

Enfin, je voudrais remercier tous ceux qui m'ont supporté pendant mes études, en particulier les ex-lyonnais, Bruno, Guilhem, Florent, Étienne, Alexis, ainsi que les niçois, Julien, Damien et Mug.

Et pour finir ceux qui me supportent ou m'ont supporté au quotidien, ma famille, et surtout Elsa pour sa bravoure sans égale face à mon bordel constant.

Contents

Chapter 1

Introduction

Interactions between theoretical physics and computer science are often described as unilateral. Either physicists require some algorithmic knowledge to simulate some natural phenomenon, or computer scientists use different physical theories to design new computational models and study their properties. It seems, however, that the interaction between those two fields could lead to a more profound dialogue. It turns out that the current trend in theoretical physics is to identify physical systems to the 'information' they carry, and not by the 'matter' they are constituted of. In fact, this vision can be directly traced back to the postulates of different physical theories. For instance, the notion of entropy, present in thermodynamics, captures the notion of "quantity of information" inside a closed physical system. There is a growing opinion that physics will soon enter a 'computational' trend. Taking this view to an extreme in this direction, one could imagine making the *Gedankenexperiment* that the universe is nothing but a large parallel computer, processing bits of information rather than mechanical processes: this is the "digital physics" paradigm. This vision was proposed by numerous great researchers, such as Fredkin, Toffoli and Margolus [Fre92, FT82, Mar88, Mar84].

One of the most famous computation model born thanks to this approach of physics is the model of cellular automata. This model, introduced by von Neumann and Ulam in the forties, is the simplest discrete model capturing three fundamental symmetries of physics: causality (information travels with a bounded speed across space), homogeneity (the laws of physics are the same everywhere, at every time) and bounded density of information (a bounded volume of space cannot contain an unbounded amount of information). In their primary definition, cellular automata consist in arrays of cells, each of them containing a state in a finite set. The arrays evolve synchronously and in discrete time steps, the state of each cell being updated according to the states of its neighbours. In a more computer science oriented-vision, cellular automata can be seen as the synchronous application of a local rule throughout the grid. Later, in 1969, Hedlund, together with Curtis and Lyndon, gave a mathematical reality to the model by proving that cellular automata defined using local rules are, in fact, uniformly continuous and shift-commuting transformations over a set of configurations. This plurality of the formalisms places this model directly at the interface between physics, mathematics and computer science. However, the rigidity of the underlying space is often cited as a limitation of the model. Indeed, while they are the perfect choice when considering dynamics on Euclidean space, they fail to model dynamics over irregular topologies, like for instance general discrete metric spaces, or arbitrary interaction agent networks. Cellular automata have been generalized in many flavours. Some generalizations explore the way a cellular automaton can act upon its set of configurations, such as asynchronous cellular automata, while others try to generalize the configurations space. For instance, it is possible to nicely generalize cellular automata to act upon Cayley graphs. A Cayley graph is a graph representing a finitely generated group: each vertex of the graph represents an element of the group, and for each relation $a \cdot b = c$ where b is a generator, an edge between vertices a and c is added, labelled with b at one end and b^{-1} at the other. These graphs have the nice feature of being very regular: all vertices have the same degree and are indistinguishable, making it easy to generalize the notion of cellular automata acting upon them. Even though this generalization allows the definition of cellular automata acting upon a much relaxed set of configuration, the underlying space is still regular and fixed in time.

However, there are many situations where the interaction of some agents (for instance some particles, some computer process, or some biological agent) according to some interactions networks, leads to a change in the network itself. A good example would be a network composed of people and their address books. The interaction network would change locally (e.g. you could share your contacts with one of your friend), or maybe a new person could enter the network. Another, more physical, example would the case of general relativity. In this case, the network would represent space itself, with its curvature. Some massive particle will then move across space along the path dictated by its curvature. The motion of the particle will, in turn, modify space curvature, thus modifying the underlying network. Modelling those dynamics require a less restricted, time-varying interaction network. There have been several approaches to generalize cellular automata not just to Cayley graphs, but to arbitrary connected graphs of bounded degree, with, sometimes, time-variation of the topology:

- With a fixed, arbitrary topology, in order to describe certain distributed algorithms [PR02, DMG08, Gru10], or to generalize the Garden-of-Eden theorem [Gro99, CSS04].
- Through the simulation environments of [GS08, VMPDJ11, KKBS05] which offer the possibility of applying a local rewriting rule simultaneously in different non-conflicting places.
- Through concrete instances advocating the concept of cellular automata extended to time-varying graphs as in [TKM09, KK07, KCN$^+$10], some of which are advanced algorithmic constructions [TKM02, TMKK05].
- Through amalgamated graph transformations [BFH87, Löw93] and parallel graph transformations [EL93, Tae96, Tae97], which work out rigorous ways to apply a local rewriting rule synchronously throughout a graph.
- Through rule-based graph rewriting languages [DL04], or their stochastic versions [DFF$^+$12].

The approach of this thesis is different in the sense that it first generalizes Cayley graphs and then applies the mathematical characterization of cellular automata as the set of shift-invariant continuous transformations in order to generalize cellular automata, thus defining a new model: *Causal Graph Dynamics*. Compared to the above cellular automata approaches, our model extends the fundamental structure theorems about cellular automata to arbitrary, connected, bounded degree, time-varying graphs. Compared with the above mentioned graph rewriting papers, the contribution is to deduce aspects of amalgamated/parallel graph transformations from the axiomatic and topological properties of the global function.

The subject of this thesis is to study and develop this model, which was first introduced by Arrighi and Dowek in [AD12]. In this first version, the authors already achieved an extension of cellular automata to arbitrary, bounded degree, time-varying graphs, although through a notion of continuity, with the same motivations. However, this work failed to endow the configuration space with a compact metric, necessary to the formulation of a similar result as Hedlund's theorem, and the existence of such a metric was left as an open question. It also leaves open whether causal graph dynamics are (locally) computable. Indeed, in this early formalization of the model, the graph structure used to define the configuration space of the cellular automata had identified vertices. The shift-invariance of a graph transformation was then naturally translated as a weak commutation with renamings of the vertices. While this allows a relatively compact definition for shift-invariance, it does not prevent the graph transformation to hide possibly uncomputable functions inside the identifiers of the vertices. In order to tackle this issue, we had to define a new model of graph, where vertices have no identifier whatsoever.

This thesis is organized as follows. Chapter 2 is dedicated to the construction of this model of graph. The introduction of this new model of graph allows to define our model of causal graph dynamics as graph transformations satisfying three properties: continuity, shift-invariance and boundedness. Alternatively, these transformations can also be defined as the synchronous mapping of a local rule on every vertex of a graph (see Figure 1.1).

These two definitions and their equivalence are presented in Chapter 3, thus generalizing Hedlund's theorem. We will then focus on an additional symmetry almost omnipresent in physics: reversibility. Given the variety of results in reversible cellular automata theory, it seems quite natural to study this additional property when considering extension of cellular automata. More particularly, the fact that causal graph dynamics can, for instance, change the number of vertices of a graph brings novel questions. Another, more physical, motivation is the design of a quantum version of causal graph dynamics, which is yet to achieve. Several aspects of this symmetry are addressed in chapter 4. First, the correspondence between invertibility and reversibility is studied. Then, a decomposition of reversible causal graph dynamics into bounded-depth circuit of reversible local gates is presented. Another fructifying area of discrete computational models is the study of their universality. Intrinsic universality, in particular, is a subject of choice. This type of universality explores the capacity of the model to simulate itself efficiently, and has been intensively studied for cellular automata and their various extensions. In the scope of studying a computational model as a physical toy model, the quest of the simplest universal instance of the model takes a completely new meaning, and could help us understand the true structure of physical laws. Another, less common, form of universality

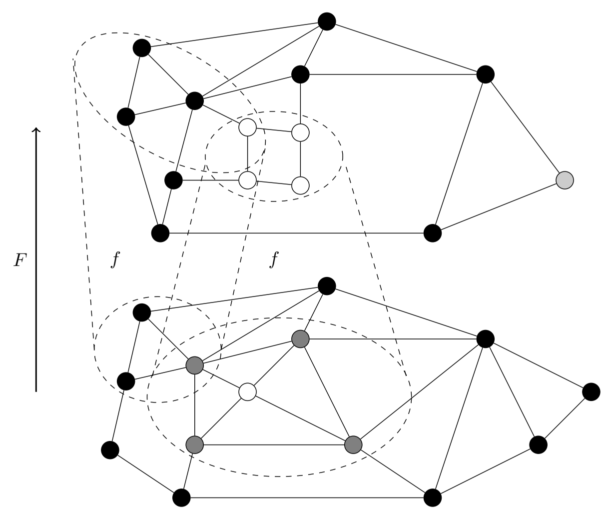

Figure 1.1: Representation of a graph transformation F induced by the homogeneous and synchronous application of a local rule f on every vertex of a graph.

is von Neumann's notion of universal construction machine, where a machine can read a description of any instance of the model and build this instance. Chapter 5 provides a natural definition of intrinsic universality and describes the construction of a family of intrinsically universal local rules, together with a universal construction machine. A natural application of causal graph dynamics to physics is the modelling of discrete version of general relativity. Even though this model seems adequate, in terms of structure of the dynamics, to the study of this complicated field, the structure of the configuration space is still too relaxed to be considered as a discrete geometrical space. Chapter 6 explores a correspondence between generalized Cayley graphs and Δ-complexes which leads, in the two-dimensional case, to a definition of discrete surface causal dynamics.

How to read this thesis?

Even though the origin of causal graph dynamics model lies in physics, I was conducted to study very different aspects of this model, some of which have nothing to do with constructing a discrete model able to embed the power of relativity. Here are some guidelines to read this thesis. Although Chapter 2 and 3 are mandatory to get the basic definitions of the model, a reader having no particular interest in the axiomatic approach of the model might want to focus its reading on sections 2.1, and 3.2, which provides sufficient background to understand the universal construction of Chapter 5. On the other hand, a reader whose interest lies in the reversibility of CGD, and in particular the relations between invertibility and reversibility, might want to skip the details of the local rule construction, as it is not a requirement for the results of Chapter 4.

Chapter 2

Generalized cayley graphs

Classical graphs. In the original work of Arrighi and Dowek [AD12], the model of causal graph dynamics was based on what we could refer to as "usual" graphs, with the added feature of using ports to name the neighbours of a vertex and having a bound on the degree of the graph. This model is sufficient to define cellular automata over time-varying graphs. In order to define local rules over graphs, the authors needed to use identifiers in vertices. The influence of those identifiers was then limited by enforcing a weakened commutation relation between the cellular automaton and any renaming of the vertices. Even though the intricacies of these identifiers was sort of decoupled from the rest of the evolution, this still made it impossible to prove that applying a causal graph dynamics to a finite graph was a computable process, which seemed to be a desirable property. It became natural to try to embed this absence of influence of the identifiers, by just removing these identifiers right from the start in the graph model itself, rather than "artificially" removing them when defining the cellular automata.

Cayley graphs. Cayley graphs are directed graphs, or digraphs, encoding a finitely generated group. The Cayley graph of a group having a set of generators $S = \{s_1, s_1^{-1}, ..., s_n, s_n^{-1}\}$ is a graph whose vertices are the elements of the group and whose edges are of the form $(g, g \cdot s)$ where $s \in S$ and $\cdot$ denotes the group operation. The regularity of these graphs allows to easily generalize the classical definition of cellular automata by replacing the $n-$dimensional Euclidean grid topology by a Cayley graph. One interesting feature of these graphs is the fact that, even though their vertices are labelled by an element of the group, one does not need this information to refer to a precise vertex. Indeed one could imagine referring to a vertex by using the sequence of edges we need to traverse to reach it, starting from the identity. To go even further, we might want to fully describe the target vertex, and give the set of all paths starting from the identity and leading to it. In fact this representation is equivalent to describe the group as a language over he set of its generators S considered as a finite alphabet, together with an equivalence relation induced by the group equality.

Generalized Cayley graphs. Now consider a "classical" graph where a vertex is pointed. Even though it is not a Cayley graph, we can still describe a vertex of this graph by using the set of paths starting from the pointed vertex and leading to it. Using ports to number the adjacent edges of the vertices, each path can be seen as a word over the alphabet of the ports and the set of all paths as a language. We can still quotient this language by the relation "lead to the same vertex". The corresponding structure is not necessarily a group, but can represent any "classical" graph. This structure is what we call a *generalized Cayley graph*.

The content of this chapter is based on [AMN13] and [AM12], co-authored with Pablo Arrighi and Vincent Nesme.

2.1 Brief overview.

This first section offers a brief overview of the model of generalized Cayley graph (or pointed graph modulo) used in this thesis. Sections 2.2, 2.3 and 2.4 provide the formal definition of the graph model and its properties. Finally, appendix A provides all the details and interpretations of this model.

Pointed graph modulo. Basically, the pointed graphs modulo are the usual, connected, undirected, possibly infinite, bounded-degree graphs, but with a few added twists:

- Each vertex has *ports* in a finite set π. A vertex and its port are written $u\!:\!a$.
- An *edge* is an unordered pair $\{u\!:\!a, v\!:\!b\}$. I.e. edges are between ports of vertices, rather than vertices themselves. Because the port of a vertex can only appear in one edge, the degree of the graphs is bounded by $|\pi|$. We shall consider connected graphs only.

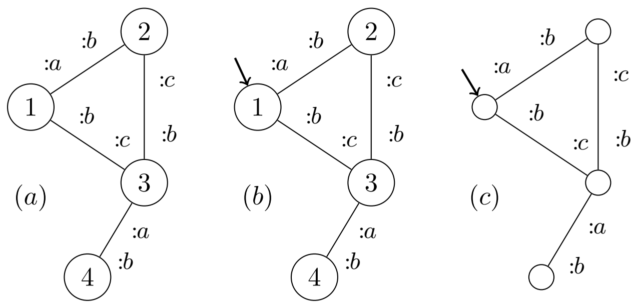

Figure 2.1: *The different types of graphs.* (a) A graph. (b) A pointed graph. (c) A generalized Cayley graph. In (c), vertices have no name and the formal way of describing this graph structure is given in section A.

- There is a privileged pointed vertex playing the role of an origin, so that any vertex can be referred to relative to the origin, via a sequence of ports that lead to it.
- The graphs are considered modulo isomorphism, so that only the relative position of the vertices can matter.
- The vertices and edges are given labels taken in finite sets Σ and Δ, so that they may carry an internal state just like the cells of a cellular automaton.
- The labelling functions are partial, so that we may express our partial knowledge about part of a graph. For instance it is common that a local function may yield a vertex, its internal state, its neighbours, and yet have no opinion about the internal state of those neighbours.

The set of all pointed graphs modulo (see Figure 2.1(c)) of ports π, vertex labels Σ and edge labels Δ is denoted $\mathcal{X}_{\Sigma,\Delta,\pi}$.

Paths and vertices. Since we are considering pointed graphs modulo isomorphism, vertices no longer have a unique identifier, which may seem impractical when it comes to designating a vertex. Two elements come to our rescue. First, these graphs are pointed, thereby providing an origin. Second, the vertices are connected through ports, so that each vertex can tell between its different neighbours. It follows that any vertex of the graph can be designated by a sequence of ports in $(\pi^2)^*$ that lead from the origin to this vertex. The origin is designated by ε. For instance, say two vertices designated by a path u and a path v, respectively. Suppose there is an edge $e = \{u : a, v : b\}$. Then, v can be designated by the path $u.ab$, where "." stands for the word concatenation.

Operations. Given a pointed graph modulo X, X^r denotes the subdisk of radius r around the pointer. The pointer of X can be moved along a path u, leading to $Y = X_u$. The pointer can be moved back where it was before, leading to $X = Y_{\overline{u}}$. We use the notation X_u^r for $(X_u)^r$ i.e., first the pointer is moved along u, then the subdisk of radius r is taken. Figure 2.2 describes those two operations.

2.2 Generalized Cayley Graphs

The current section formalizes the notion of generalized Cayley graphs. Fundamental algebraic properties, in terms of languages and comparison with Cayley graphs, are provided in Appendix A.

Notations. Let π be a finite set, $\Pi = \pi^2$ denotes pairs of elements in π, and $V = \mathcal{P}(\Pi^*)$ the set of languages over the alphabet Π. The operator '.' represents the concatenation of words and ε the empty word, as usual.

Whilst generalized Cayley graphs will be up to isomorphism, we still need to manipulate plain graphs, non-modulo, at different stages. The *vertices* of these graphs (See Figure 2.1(a)) we consider in this work are uniquely identified by a name u in V. (This particular choice of the universe of names is actually irrelevant until Definition 8, where it becomes natural.) A vertex u may also be labelled with a *state* $\sigma(u)$ in Σ a finite set. Each vertex has *ports* in the finite set π. A vertex u and its port a is denoted $u:a$.
An *edge* is an unordered pair $\{u:a, v:b\}$. Such an edge connects vertices u and v; We shall consider connected graphs only. The port of a vertex can only appear in one edge, so that the degree of the graphs is always bounded by $|\pi|$. Edges may also be labelled with a *state* $\delta(\{u:a, v:b\})$ in Δ a finite set.
Definitions 1 to 4 are as in [AD12]. The first two are reminiscent of the many papers seeking to generalize cellular automata to arbitrary, bounded degree, fixed graphs [PR02, DMG08, Gru10, Gro99, CSS04, TKM09, KK07, TKM02,

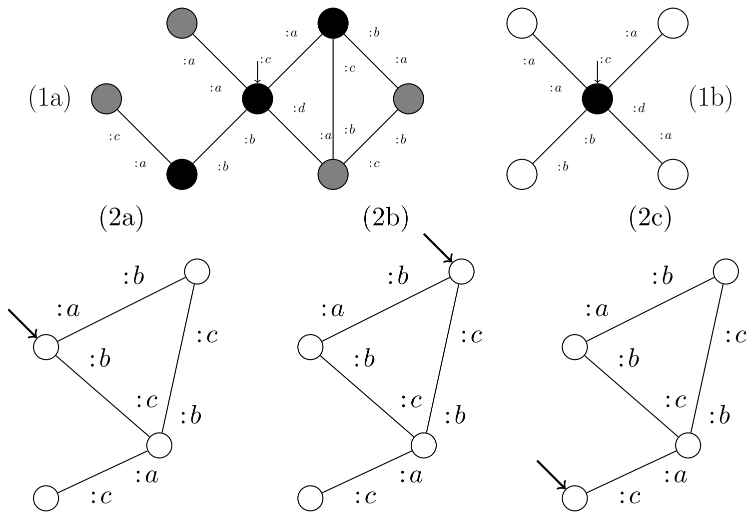

Figure 2.2: *Operations over pointed graphs modulo.* (1) From X to X^0: taking the *subdisk of radius* 0. In general the neighbours of radius r are just those vertices which can be reached in r steps starting from the origin, whereas the disk of radius r, written X^r, is the subgraph induced by the neighbours of radius $r+1$, with labellings restricted to the neighbours of radius r and the edges between them. (2a) A pointed graph modulo X. (2b) X_{ab} the pointed graph modulo X *shifted* by ab. (2c) $X_{bc.ac}$ the pointed graph modulo X *shifted* by $bc.ac$, which also corresponds to the graph X_{ab} *shifted* by $cb.ac$. *Shifting* this last graph by $\overline{cb.ac} = ca.bc$ produces the graph (2b) again.

TMKK05, BFH87, Löw93, EL93, Tae96, Tae97]. They are illustrated by Figure 2.1(a).

Definition 1 (Graph). *A* graph *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* $\forall e, e' \in E(G), e \cap e' \neq \varnothing \Rightarrow e = e'$.

The graph is assumed to be connected: for any two $u, v \in V(G)$, *there exists* $v_0, \ldots, v_n \in V(G)$, $a_0, b_0 \ldots, a_{n-1}, b_{n-1} \in \pi$ *such that for all* $i \in \{0 \ldots n-1\}$, *one has* $\{v_i:a_i, v_{i+1}:b_i\} \in E(G)$ *with* $v_0 = u$ *and* $v_n = v$.

Definition 2 (Labelled graph). *A labelled graph is a triple* (G, σ, δ), *also denoted simply* G *when it is unambiguous, where* G *is a graph, and* σ *and* δ *respectively label the vertices and the edges of* G*:*

- σ *is a partial function from* $V(G)$ *to a finite set* Σ*;*
- δ *is a partial function from* $E(G)$ *to a finite set* Δ.

The set of all graphs *with ports* π *is written* $\mathcal{G}_\pi$. *The* set of labelled graphs *with states* Σ, Δ *and ports* π *is written* $\mathcal{G}_{\Sigma,\Delta,\pi}$. *To ease notations, we sometimes write* $v \in G$ *for* $v \in V(G)$.

In definition 2 the labelling functions are possibly partial, e.g. a vertex may be potentially stateless. Allowing for this possibility is convenient to describe local rules, which produce vertices and their relations to neighbour vertices, without necessarily having an opinion on the states of the neighbour vertices. A concrete example of this is given in Section 3.2 and Figure 3.4.

We now want to single out a vertex. Definition 3 is illustrated by Figure 2.1(b).

Definition 3 (Pointed graph). *A* pointed (labelled) graph *is a pair* (G, p) *with* $p \in G$. *The* set of pointed graphs *with ports* π *is written* $\mathcal{P}_\pi$. *The* set of pointed labelled graphs *with states* Σ, Δ *and ports* π *is written* $\mathcal{P}_{\Sigma,\Delta,\pi}$.

The idea is now to get rid of all the unnecessary information in these graphs. Definition 4 of isomorphism formalizes the notion of vertex renaming in a graph.

Definition 4 (Isomorphism). *An* isomorphism R *is a function from* $\mathcal{G}_\pi$ *to* $\mathcal{G}_\pi$ *which is specified by a bijection* $R(.)$ *from* V *to* V. *The image of a graph* G *under the isomorphism* R *is a graph* RG *whose set of vertices is* $R(V(G))$, *and whose set of edges is* $\{\{R(u) : a, R(v) : b\} \mid \{u : a, v : b\} \in E(G)\}$. *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. The definition extends to pointed labelled graphs.*

Notice that pointed graph isomorphism renames the pointer in the same way as it renames the vertex upon which it points; which effectively means that the pointer does not move. Later we shall introduce a distinct kind of operation, which moves the pointer, not to be confused with this isomorphism. When describing a graph, we do not need to specify the name or the identity of the vertices in order to uniquely describe this graph. In definition 5, we use the notion of isomorphism to get rid of all names in the graph.

Definition 5 (Generalized Cayley graphs). *Let P be a pointed (labelled) graph (G, p). The* generalized Cayley graph *X is $\widetilde{P}$ the equivalence class of P with respect to the equivalence relation $\approx$. The* set of generalized Cayley graphs *with ports π is written $\mathcal{X}_\pi$. The* set of labelled generalized Cayley graphs *with states Σ, Δ and ports π is written $\mathcal{X}_{\Sigma,\Delta,\pi}$.*

These pointed graphs modulo will constitute the set of *configurations* of the generalized cellular automata that we will consider in this work.

We will need a notion of path in a generalized Cayley graph:

Definition 6 (Path). *Given a generalized Cayley graph 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 traverse the graph according to this sequence. More formally, α is a path if and only if there exists $(G, p) \in X$ and there also exists $v_0, \dots, v_n \in V(G)$ such that for all $i \in \{0, \dots, n-1\}$, one has $\{v_i : a_i, v_{i+1} : b_i\} \in E(G)$, with $v_0 = p$ and $\alpha_i = a_i b_i$. Notice that the existence of a path does not depend on the choice of $(G, p) \in X$. The set of paths of X is denoted by $L(X)$.*

Notice that 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 $\overline{\alpha}$ which stands for mirror of path α. The detailed algebraic structure of the set of paths $L(X)$ of a generalized Cayley graph X is described in Appendix A.

Two paths are equivalent if they lead to same vertex:

Definition 7 (Equivalence of paths). *Given a generalized Cayley graph*

X, we define the equivalence of paths relation, denoted $\equiv_X$, on $L(X)$ such that for all paths $\alpha, \alpha' \in L(X)$, $\alpha \equiv_X \alpha'$ if and only if, starting from the pointer, α and α' lead to the same vertex of X. More formally, $\alpha \equiv_X \alpha'$ if and only if there exists $(G,p) \in X$ and $v_1, \ldots, v_n, v'_1, \ldots, v'_{n'} \in V(G)$ such that for all $i \in \{0 \ldots n-1\}$, $i' \in \{0 \ldots n'-1\}$, one has $\{v_i : a_i, v_{i+1} : b_i\} \in E(G)$, $\{v'_{i'} : a'_{i'}, v'_{i'+1} : b'_{i'}\} \in E(G)$, with $v_0 = p$, $v'_0 = p$, $\alpha = (a_i b_i)_{i \in \{0,\ldots,n-1\}}$, $\alpha' = (a'_{i'} b'_{i'})_{i \in \{0,\ldots,n'-1\}}$ and $v_n = v_{n'}$. We write $\tilde{\alpha}$ for the equivalence class of α with respect to $\equiv_X$.

For mainly technical reasons, it will often be useful to undo the modulo, i.e. to obtain a canonical instance of a pointed graph modulo.

Definition 8 (Associated graph). *Let X be a generalized Cayley graph. Let $G(X)$ be the graph such that:*

- *The set of vertices $V(G(X))$ is the set of equivalence classes of $L(X)$;*
- *The edge $\{\tilde{\alpha} : a, \tilde{\beta} : b\}$ is in $E(G(X))$ if and only if $\alpha.ab \in L(X)$ and $\alpha.ab \equiv_X \beta$, for all $\alpha \in \tilde{\alpha}$ and $\beta \in \tilde{\beta}$.*

We define the associated graph *to be $G(X)$.*

Conventions. Appendix A proves that:

- a generalized Cayley graph X,
- its associated graph $G(X)$
- the algebraic structure $\langle L(X), \equiv_X \rangle$

can be viewed as three presentations of the same mathematical object. It further provides an axiomatization of these algebraic structures. Altogether, this justifies the fact that each vertex of this mathematical object can be designated by

- $\tilde{\alpha}$ an equivalence class of $L(X)$, i.e. the set of all paths leading to this vertex starting from $\tilde{\varepsilon}$,
- or more directly by α an element of an equivalence class $\tilde{\alpha}$ of X, i.e. a particular path leading to this vertex starting from ε.

These two remarks lead to the following mathematical conventions, which we adopt for convenience. From now on:

- $\tilde{\alpha}$, α will no longer be distinguished. The latter notation will be given the meaning of the former. We shall speak of a "vertex" α in $V(X)$ (or simply $\alpha \in X$).
- It follows that '$\equiv_X$' and '$=$' will no longer be distinguished. The latter notation will be given the meaning of the former. I.e. we shall speak of "equality of vertices" $\alpha = \beta$ (when strictly speaking we just have $\tilde{\alpha} = \tilde{\beta}$).

In any case, we will make sure that a rigorous meaning can always be recovered by placing tildes back.

2.3 Basic operations

2.3.1 Operations on generalized Cayley graphs

For a pointed graph (G, p) non-modulo (see [AD12] for details):

- the neighbours 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 neighbours of radius $r+1$, with labellings restricted to the neighbours of radius r and the edges between them, and pointed at p.

Notice that the vertices of G_p^r continue to have the same names as they used to have in G. For generalized Cayley graphs, on the other hand, the analogous operation is:

Definition 9 (Disk). *Let $X \in \mathcal{X}_{\Sigma,\Delta,\pi}$ be a generalized Cayley graph and G its associated graph. Let X^r be $\widetilde{G_\varepsilon^r}$. The generalized Cayley graph $X^r \in \mathcal{X}_{\Sigma,\Delta,\pi}$ is referred to as the* disk of radius r *of X. The* set of disks of radius r *with states Σ, Δ and ports π is written $\mathcal{X}_{\Sigma,\Delta,\pi}^r$.*

A technical remark is that the vertices of X^r no longer have quite the same names as they used to have in X. This is because, in a generalized Cayley graph, vertices are designated by those paths that lead to them, starting from the vertex ε, and there were many more such paths in X than there are in its subgraph X^r. Still, it is clear that there is a natural inclusion $V(X^r) \subseteq V(X)$, meaning that $u \in X^r$ implies that there exists a unique $u' \in X$ such that

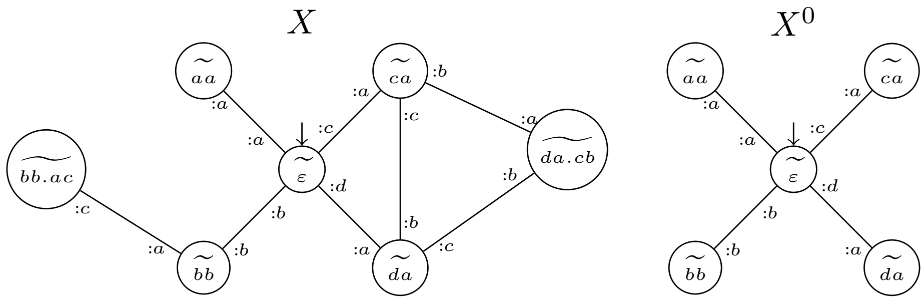

Figure 2.3: *A generalized Cayley graph and its disk of radius* 0. Notice that the equivalence classes describing vertices in X^0 are strict subsets of those in X, even though their shortest representative is the same. For instance the path $ca.cb$ is in $\tilde{da}$ in X but is not a path in X^0, and thus does not belong to $\tilde{da}$ in X^0.

$u \subseteq u'$. Thus, we will commonly say that a vertex of $u \in X^r$ belongs to X, even though technically we are referring to the corresponding vertex u' of X. Similarly, we will commonly say that a vertex of $u' \in X$ belongs to X^r when we actually mean that there is a unique vertex u of X^r such that $u \subseteq u'$.

Definition 10 (Size). *Let $X \in \mathcal{X}_{\Sigma,\Delta,\pi}$ be a generalized Cayley graph. We say that a vertex $u \in X$ has size less or equal to $r+1$, and write $|u| \leq r+1$, if and only if $u \in X^r$. We denote $V(\mathcal{X}^r_\pi) = \bigcup_{X \in \mathcal{X}^r_\pi} V(X)$.*

It will help to have a notation for the graph where vertices are named relatively to some other pointer vertex u.

Definition 11 (Shift). *Let $X \in \mathcal{X}_{\Sigma,\Delta,\pi}$ be a generalized Cayley graph and G its associated graph. Consider $u \in X$ 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 $\widetilde{(G,u)}$. The generalized Cayley graph X_u is referred to as X* shifted by *u.*

The composition of a shift, and *then* a restriction, applied on X, will simply be written X^r_u. Whilst this is the analogous operation to G^r_u over pointed graphs non-modulo, notice that the shift-by-u completely changes the names of the vertices of X^r_u. As the naming has become relative to u, the disk X^r_u

holds no information about its prior location, u.
We may also want to designate a vertex v by those paths that lead to the vertex u relative to ε, followed by those paths that lead to v relative to u. The following definition of concatenation coincides with the one that is induced by the concatenation of words belonging to the classes u and v:

Definition 12 (Concatenation). *Let $X \in \mathcal{X}_\pi$ be a generalized Cayley graph and G its associated graph. Consider $u \in X$ and $v \in X_u$ or X_u^r for some r. Let G' be the associated graph of $(X_u)_v$, R be an isomorphism such that $G' = RG$, and $u.v$ be $R^{-1}(\varepsilon)$. The vertex $u.v \in X$ is referred to as* u concatenated with v.

According to Definition 11, G' and G are isomorphic. Moreover, the restriction of R^{-1} to $V(G')$ is uniquely determined; hence definition 12 is sound.
It also helps to have a notation for the paths to ε relative to u.

Definition 13 (Inverse). *Let $X \in \mathcal{X}_\pi$ be a generalized Cayley graph and G its associated graph. Consider $u \in X$. Let G' be the associated graph of X_u and R be an isomorphism such that $G' = RG$, and $\overline{u}$ be $R(\varepsilon)$. The vertex $\overline{u} \in X_u$ is referred to as the* inverse of u.

Notice the following easy facts: $(X_u)_v = X_{u.v}$, $u.\overline{u} = \varepsilon$. Notice also that the isomorphism R such that $G(X_u) = RG(X)$ maps v to $\overline{u}.v$. This last property suggests that we may define shifts upon graphs (non-modulo) as a certain class of isomorphisms. In order to formalize this notion within the set of graphs without appealing to graphs modulo, we will need that the vertices of our graphs non-modulo be of a particular form.

2.3.2 Operations on graphs

In Section 2.2 we said that a graph $G \in \mathcal{G}_\pi$ would have vertex names in V. But now we shall allow vertices to have names in disjoint subsets of $V.S$, with $S = \{\varepsilon, 1, 2, \ldots, b\}$ a finite set of suffixes. For instance, given some generalized Cayley graph X, having vertices u, v in $V(X)$, we may build some graph G having vertices $\{v\}$, $\{u.1\}$, $\{u.3, v.1\}$ ...i.e. subsets of $V(X).S$. Later, $\{u.1\}$ will be interpreted as the vertex which is 'the first successor of u', $\{u.3, v.1\}$ as the vertex which is 'the first successor of v and the third successor of u', $\{v\}$ as the vertex which is 'the continuation of v'. Disjointness is just to keep things tidy: one cannot have a vertex which is the first successor of u ($\{u.1\}$, say) coexisting with another which is the 'the first successor of u and the second successor of v' ($\{u.1, v.2\}$, say) — although some other convention could have been used. Still, some form of suffixes is necessary in order to provide just the little, extra naming space that is needed in order to create new vertices. Figure 2.4 illustrates this restriction over the vertices names.

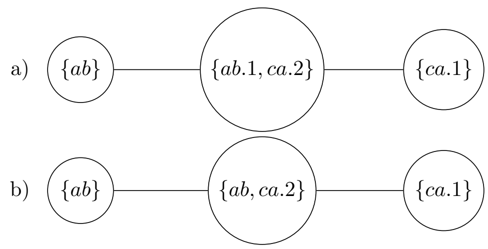

Figure 2.4: a) Is a valid graph as all its vertices names are disjoint subsets. However, b) is not valid as vertices names $\{ab\}$ and $\{ab, ca.2\}$ intersect.

Definition 14 (Shift isomorphism). *Let $X \in \mathcal{X}_\pi$ be a generalized Cayley graph. Let $G \in \mathcal{G}_\pi$ be a graph that has vertices that are disjoint subsets of $V(X).S$ or $V(X^r).S$ for some r. Consider $u \in X$. Let R be the isomorphism from $V(X).S$ to $V(X_u).S$ mapping $v.z \mapsto \overline{u}.v.z$, for any $v \in V(X)$ or $V(X^r)$, $z \in S$. Extend this bijection pointwise to act over subsets of $V(X).S$, and let $\overline{u}.G$ to be RG. The graph $\overline{u}.G$ has vertices that are disjoint subsets of $V(X_u).S$, it is referred to as G* shifted by u. *The definition extends to labelled graphs.*

Definitions 15 and 16 are standard, see [BFH87, Löw93] and [AD12], although here again the vertices of G are given names in disjoint subsets of $V(X).S$ for some X. Basically, we need a notion of *union* of graphs, and for this purpose we need a notion of *consistency* between the operands of the union:

Definition 15 (Consistency). *Let $X \in \mathcal{X}_\pi$ be a generalized Cayley graph. Let G be a labelled graph (G, σ, δ), and G' be a labelled graph (G', σ', δ'), each one having vertices that are pairwise disjoint subsets of $V(X).S$. The graphs are said to be* consistent *if and only if:*

(i) $\forall x \in G \, \forall x' \in G' \quad x \cap x' \neq \varnothing \Rightarrow x = x'$,

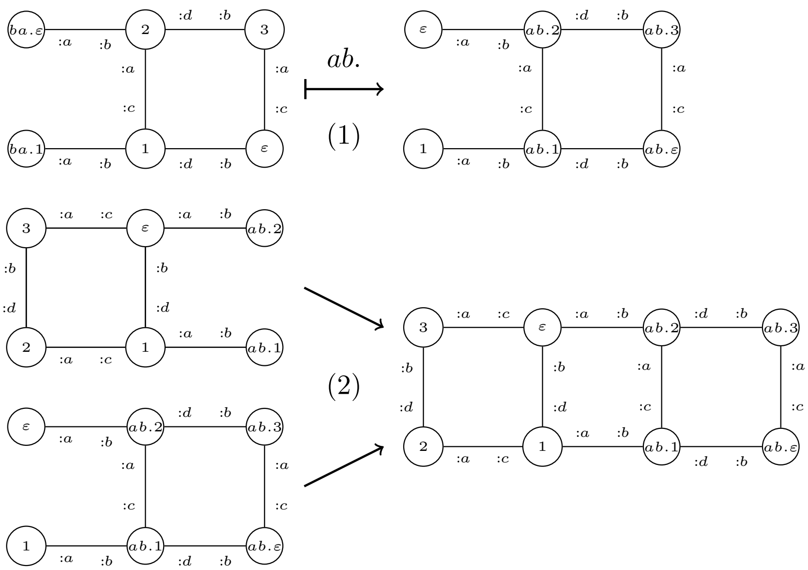

Figure 2.5: Operations over graphs. (1) A shift of a graph on the vertex ba. The structure of the graph is preserved, only the names of the vertices are changed. The new vertex ε is the former vertex ba. (2) A graph union. Here the two graphs on the left hand side intersect on vertices ε, 1, $ab.1$ and $ab.2$. As the two are consistent (e.g. in both graph, vertices ε and $ab.2$ are connected along an ab edge) their union can be computed, resulting in the right hand side graph.

(ii) $\forall x, y \in G\, \forall x', y' \in G'\, \forall a, a', b, b' \in \pi$
$(\{x\!:\!a, y\!:\!b\} \in E(G) \wedge \{x'\!:\!a', y'\!:\!b'\} \in E(G') \wedge x = x' \wedge a = a') \Rightarrow (b = b' \wedge y = y')$,

(iii) $\forall x, y \in G\, \forall x', y' \in G'\, \forall a, b \in \pi \quad x = x' \Rightarrow \delta(\{x\!:\!a, y\!:\!b\}) = \delta'(\{x'\!:\!a, y'\!:\!b\})$ *when both are defined,*

(iv) $\forall x \in G\, \forall x' \in G' \quad x = x' \Rightarrow \sigma(x) = \sigma'(x')$ *when both are defined.*

They are said to be trivially consistent *if and only if for all $x \in G$, $x' \in G'$ we have $x \cap x' = \varnothing$.*

The consistency conditions aim at making sure that both graphs "do not disagree". Indeed: (iv) means that "if G says that vertex x has label $\sigma(x)$, G' should either agree or have no label for x"; (iii) means that "if G says

that edge e has label $\delta(e)$, G' should either agree or have no label for e"; (ii) means that "if G says that starting from vertex x and following port a leads to y via port b, G' should either agree or have no edge on port $x\!:\!a$".
Condition (i) is in the same spirit: it requires that G and G', if they have a vertex in common, then they must fully agree on its name. Remember that vertices of G and G' are disjoint subsets of $V(X).S$. If one wishes to take the union of G and G', one has to enforce that the vertex names will still be disjoint subsets of $V(X).S$.
Trivial consistency arises when G and G' have no vertex in common: thus, they cannot disagree on any of the above.

Definition 16 (Union). *Let $X \in \mathcal{X}_\pi$ be a generalized Cayley graph. Let G be a labelled graph (G, σ, δ), and G' be a labelled graph (G', σ', δ'), each one having vertices that are pairwise disjoint subsets of $V(X).S$. Whenever they are consistent, their* union *is defined. The resulting graph $G \cup G'$ is the labelled graph with vertices $V(G) \cup V(G')$, edges $E(G) \cup E(G')$, labels that are the union of the labels of G and G'.*

Finally, recall that for a pointed graph (G, p) non-modulo G^r_p, is the subgraph induced by the neighbours of radius $r+1$, with labellings restricted to the neighbours of radius r and the edges between them, and pointed at p [AD12].

2.4 Topological properties

Having a well-defined notion of disks allows us to define a topology upon $\mathcal{X}_{\Sigma,\Delta,\pi}$, which is the natural generalization of the well-studied Cantor metric upon cellular automata configurations [Hed69].

Definition 17 (Gromov-Hausdorff-Cantor metrics). *Consider the function*

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

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.

Soundness:
[**Nonnegativity, symmetry, identity of indiscernibles**] are obvious.
[**Equivalence**]

$$\begin{aligned} d(X,Y) < \epsilon &\Leftrightarrow d(X,Y) = 1/2^k \text{ with } k \in \mathbb{N} \wedge 1/2^k < \epsilon \\ &\Leftrightarrow k = \min\{r \in \mathbb{N} \mid X^r \neq Y^r\} \wedge 1/2^k < \epsilon \\ &\Leftrightarrow_{r=k-1}\ X^r = Y^r \text{ with } r \in \mathbb{N} \wedge 1/2^{r+1} < \epsilon \\ &\Leftrightarrow X^r = Y^r \text{ with } r = \lfloor -\log_2(\epsilon) \rfloor. \end{aligned}$$

[**Ultrametricity**] Consider k such that $1/2^k = d(X,Z)$ and l such that $1/2^l = d(X,Y)$. By definition of the metric X,Z differ only after index k and X,Y differ only after index l. Suppose $k \leq l$ so that Y,Z differ only after index k. But then $d(Y,Z) = 1/2^k$ which is $d(X,Z)$.
[**Triangle inequality**] is obvious from the ultrametricity.

The fact that generalized Cayley graphs are pointed graphs modulo, i.e. the fact that they have no "vertex name degree of freedom" is key to proving the following property. Indeed, compactness crucially relies on the set being "finite-branching", meaning that the set of possible generalized Cayley graphs, as one progressively enlarges the radius of a disk, remains finite. This does not hold for usual graphs.

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

Proof. This is essentially König's Lemma. Let us consider an infinite sequence of graphs $(X(n))_{n\in\mathbb{N}}$. Because Σ and Δ are finite, and there is an infinity of elements of $(X(n))$, there must exist a graph of radius zero X^0 such that there is an infinity of elements of $(X(n))$ fulfilling $X(n)^0 = X^0$. Choose one of them to be $X(n_0)$, i.e. $X(n_0)^0 = X^0$. Now iterate: because the degree of the graph is bounded by $|\pi|$, and because Σ and Δ are finite but there is an infinity of elements of $(X(n))$ having the above property, there must exist a pointed graph of radius one X^1 such that $(X^1)^0 = X^0$ and such that there is an infinity of elements of $(X(n))$ having $X(n)^1 = X^1$. Choose one of them as $X(n_1)$, i.e. $X(n_1)^1 = X^1$. Etc. The limit is the unique graph X' having disks $X'^k = X^k$ for all k. □

Recall the difference in quantifiers between the continuity of a function F over a metric space $(\mathcal{X}, d)$:

$$\forall X \in \mathcal{X}\, \forall \epsilon > 0\, \exists \eta > 0\, \forall Y \in \mathcal{X}, \quad d(X,Y) < \eta \Rightarrow d(F(X), F(Y)) < \epsilon,$$

and its uniform continuity:

$$\forall \epsilon > 0\, \exists \eta > 0\, \forall X, Y \in \mathcal{X}, \quad d(X,Y) < \eta \Rightarrow d(F(X), F(Y)) < \epsilon.$$

Uniform continuity is the physically relevant notion, as it captures the fact that F does not propagate information too fast. In a compact setting, it is equivalent to simple continuity, which is easier to check and is the mathematically standard notion. This is the content of Heine's Theorem, a well-known result in general topology [FAP90]: given two metric spaces $\mathcal{X}$ and $\mathcal{Y}$ and $F : \mathcal{X} \longrightarrow \mathcal{Y}$ continuous, if $\mathcal{X}$ is compact, then F is uniformly continuous.

The implications of these topological notions for cellular automata were first studied in [Hed69], with self-contained elementary proofs available in [Kar11]. For cellular automata over Cayley graphs a complete reference is [CSC10]. For causal graph dynamics [AD12], these implications had to be reproven by hand, due to the lack of a clear topology in the set of graphs that was considered. Here we are able rely on the topology of generalized Cayley graphs and reuse Heine's theorem out-of-the-box, which makes the setting of generalized Cayley graphs a very attractive one in order to generalize cellular automata.

2.5 Summary of the results

We constructed a model graph, generalized Cayley graphs, having the following properties:

- neighbours of vertices are numbered, with numbering in a finite set π, hence bounding the degree of the graph,
- a particular, pointed, vertex plays the role of origin in the graph,
- vertices are named relatively to this origin.

We defined several operations over these graphs. In particular, we defined a notion of shift, which consists in moving the origin along a given path in the graph, and a notion of disk. We also endowed the set of generalized Cayley graphs with a compact metric, allowing us to define uniformly continuous transformations over this set.

Chapter 3

Causal graph dynamics

Generalizing cellular automata. Now that our configuration space is well defined, we will present our definition of cellular automata over generalized Cayley graphs, namely causal graph dynamics. The main challenge here is to preserve the two definitions of "classical" cellular automata, giving them legitimacy as both physics toy-models and computational models, and to prove their equivalence.

The first two sections of this chapter give two alternative definitions of our model: section 3.1 is based on the notions of causality and homogeneity, while section 3.2 is based on the notion of local rule. Finally, section 3.3 provides a proof of the equivalence between these two definitions.

The content of this chapter is based on [AMN13] and [AM12], co-authored with Pablo Arrighi and Vincent Nesme.

3.1 Causal dynamics

The notion of *causality* extends the known mathematical definition of cellular automata over grids and Cayley graphs. This extension will have two main features: not only the graphs become arbitrary, but they can also vary in time.

In order to define these causal dynamics, we will need to define three properties over graph transformations. The first two properties, continuity and shift-invariance, are very similar to their equivalent in CA theory, even though their expression is less immediate due to the complexity of the configuration set. The last property, boundedness, is here to prevent our dynamics to locally create an infinite number of new vertices.

The main difficulty we encountered when elaborating an axiomatic definition of causality from $\mathcal{X}_{\Sigma,\Delta,\pi}$ to $\mathcal{X}_{\Sigma,\Delta,\pi}$ (the set of generalized Cayley graphs), was the need to establish a correspondence between the vertices of $X \in \mathcal{X}_{\Sigma,\Delta,\pi}$, and those of its image by a dynamics F, $F(X)$. Indeed, on the one hand it is important to know that a given $u \in X$ has become $u' \in F(X)$, e.g. in order to express the shift-invariance $F(X_u) = F(X)_{u'}$. But on the other hand since u' is named relative to ε, its determination requires a global knowledge of X.

The following analogy provides a useful way of tackling this issue. Say that we were able to place a white stone on the vertex $u \in X$ that we wish to follow across the application of the dynamics F. Later, by observing that the white stone is found at $u' \in F(X)$, we would be able to conclude that u has become u'. This way of grasping the correspondence between an image vertex and its antecedent vertex is a local, operational notion of an observer moving across the dynamics.

Definition 18 (Dynamics). *A dynamics $(F, R_\bullet)$ is given by*

- *a function $F : \mathcal{X}_{\Sigma,\Delta,\pi} \to \mathcal{X}_{\Sigma,\Delta,\pi}$;*
- *a map $R_\bullet$, with $R_\bullet : X \mapsto R_X$ and $R_X : V(X) \to V(F(X))$.*

For all X, the function R_X can be pointwise extended to sets, i.e. $R_X : \mathcal{P}(V(X)) \to \mathcal{P}(V(F(X)))$ maps S to $R_X(S) = \{R_X(u) \mid u \in S\}$.

The intuition is that R_X indicates which vertices $\{u', v', \cdots\} = R_X(\{u, v, \cdots\}) \subseteq V(F(X))$ will end up being marked as a consequence of $\{u, v, \cdots\} \subseteq V(X)$ being marked. Now, clearly, the set $\{(X, \mathcal{P}(V(X))) \mid X \in \mathcal{X}_{\Sigma,\Delta,\pi}\}$ is isomorphic to $\mathcal{X}_{\Sigma',\Delta,\pi}$ with $\Sigma' = \Sigma \times \{0, 1\}$. Hence, we can define the function F' that maps $(X, S) \cong X' \in \mathcal{X}_{\Sigma',\Delta,\pi}$ to $(F(X), R_X(S)) \cong F'(X') \in \mathcal{X}_{\Sigma',\Delta,\pi}$, and think of a dynamics as just this function $F' : \mathcal{X}_{\Sigma',\Delta,\pi} \to \mathcal{X}_{\Sigma',\Delta,\pi}$. This alternative formalism will turn out to be very useful.

Definition 19 (Shift-invariance). *A dynamics $(F, R_\bullet)$ is said to be* shift-invariant *if and only if for every X and $u \in X$, $v \in X_u$,*

- *$F(X_u) = F(X)_{R_X(u)}$*
- *$R_X(u.v) = R_X(u).R_{X_u}(v)$.*

The second condition expresses the shift-invariance of $R_\bullet$. Notice that $R_X(\varepsilon) = R_X(\varepsilon).R_X(\varepsilon)$; hence $R_X(\varepsilon) = \varepsilon$.
In the $F' : \mathcal{X}_{\Sigma',\Delta,\pi} \to \mathcal{X}_{\Sigma',\Delta,\pi}$ formalism, the two above conditions are equivalent to just one: $F'(X_u) = F'(X)_{R_X(u)}$.

Definition 20 (Continuity). *A dynamics $(F, R_\bullet)$ is said to be* continuous *if and only if:*

- *$F : \mathcal{X}_{\Sigma,\Delta,\pi} \to \mathcal{X}_{\Sigma,\Delta,\pi}$ is continuous,*
- *For all X, for all integer m, there exists an integer n such that for all X', $X'^n = X^n$ implies $dom\, R^m_{X'} \subseteq V(X'^n)$, $dom\, R^m_X \subseteq V(X^n)$ and $R^m_{X'} = R^m_X$.*

where R^m_X denotes the partial map obtained as the restriction of R_X to the codomain $F(X)^m$, using the natural inclusion of $F(X)^m$ into $F(X)$.

The second condition expresses the continuity of $R_\bullet$. It can be reinforced into uniform continuity: for all m, there exists n such that for all X, X', $X'^n = X^n$ implies $R^m_{X'} = R^m_X$.
Indeed, in the $F' : \mathcal{X}_{\Sigma',\Delta,\pi} \to \mathcal{X}_{\Sigma',\Delta,\pi}$ formalism, the two above conditions are equivalent to just one: F' continuous. But since continuity implies uniform continuity upon the compact space $\mathcal{X}_{\Sigma',\Delta,\pi}$, it follows that F' is uniformly continuous, and thus the reinforced second condition.
We need one third, last condition:

Definition 21 (Boundedness). *A dynamics $(F, R_\bullet)$ from $\mathcal{X}_{\Sigma,\Delta,\pi}$ to $\mathcal{X}_{\Sigma,\Delta,\pi}$ is said to be* bounded *if and only if there exists a bound b such that for all X, for all $w' \in F(X)$, there exists $u' \in \operatorname{im} R_X$ and $v' \in F(X)^b_{u'}$ such that $w' = u'.v'$.*

With the help of these three conditions, we can state our main definition:

Definition 22 (Causal dynamics). *A dynamics is* causal *if it is shift-invariant, continuous and bounded.*

Example: The inflating grid. An example of causal dynamics is the inflating grid dynamics illustrated in Figure 3.1. In the inflating grid dynamics each vertex gives birth to four distinct vertices, such that the structure of the initial graph is preserved, but inflated. The graph has maximal degree

4, and the set of ports is $\pi = \{a, b, c, d\}$, vertices and edges are unlabelled. For this dynamics, the $R_\bullet$ operator is defined as follows:

$$R_X(u_0 \cdot u_1 \cdot \dots \cdot u_n) = R(u_0) \cdot R(u_1) \cdot \dots \cdot R(u_n)$$

where R is the function acting on letters in π^2 described in the following table:

$u \in \pi^2$	$R(u)$
aa	aa.db
ab	ab.db.ac
ac	ac.ac
ad	ad.bd
ba	bd.ba.db
bb	bd.bb.db.ac
bc	bd.bc.ac
bd	bd.bd
ca	ca.ca
cb	ca.cb.db
cc	ca.cc.db.ac
cd	ca.cd.ac
da	da
db	db.db
dc	dc.db.ac
dd	dd.ac

For instance, if two vertices are separated by a single edge cb in X, then moving the pointer from the first one to the second one will result in moving the pointer along the path $R(cb) = ca.cb.db$ in $F(X)$.

Lemma 2 (Bounded inflation). *Consider a causal dynamics F from $\mathcal{X}_{\Sigma,\Delta,\pi}$ to $\mathcal{X}_{\Sigma,\Delta,\pi}$. There exists a bound b such that for all X and $u \in X^r$, we have $|R_X(u)| \leq (r+1)b$.*

Proof. Let $ac \in \Pi$, and let E the subset of $\mathcal{X}_{\Sigma,\Delta,\pi}$ of those X such that $ac \in X$. E is closed — any sequence of elements of E converging in $\mathcal{X}_{\Sigma,\Delta,\pi}$ converges in E — and $\mathcal{X}_{\Sigma,\Delta,\pi}$ is compact, therefore E is compact. By continuity, the function $X \mapsto |R_X(ac)|$ is continuous from E to $\mathbb{N}$; since E is compact, it must be bounded. The result then follows from the triangle inequality and shift-invariance. $\square$

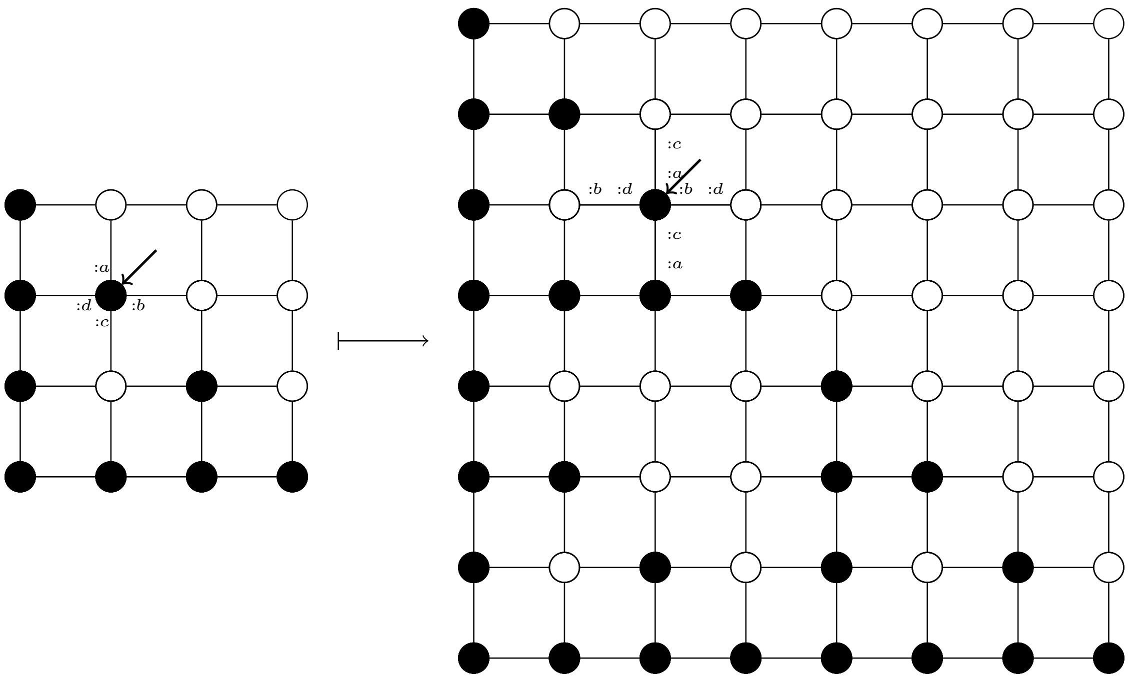

Figure 3.1: *The inflating grid dynamics.* Each vertex splits into 4 vertices. The structure of the grid is preserved. For this precise graph, all edges are connected to ports as stipulated on the pointed vertex (port $:a$ on top, $:b$ on the right, $:c$ on the bottom and $:d$ on the left).

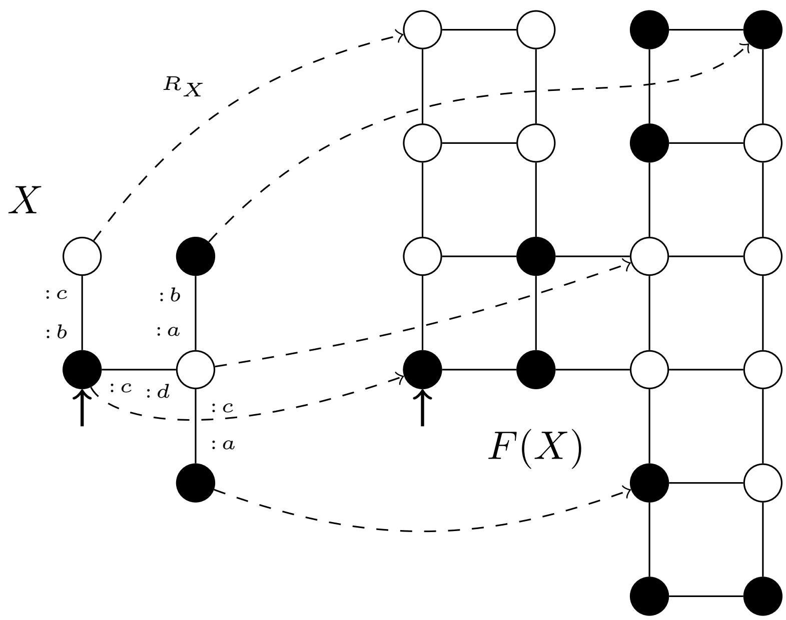

Figure 3.2: To each original vertex of X, R_X associates a vertex of $F(X)$ within the square of four it creates. More precisely, it is mapped to that of the four vertices whose ports a and d get out of the square.

3.2 Localizable dynamics

The notion of localizability of a dynamics F captures exactly the same idea as the constructive definition of a cellular automata, namely that F arises as a single local rule f applied synchronously and homogeneously across the input graph.
The general idea is that the local rule f looks at a portion of the generalized Cayley graph X (a disk X^r) and produces a piece of graph $G = f(X^r)$. The same is done synchronously at every location $u \in X$ producing pieces of graph $G' = f(X_u^r)$. The produced pieces must be consistent (see Subsection 2.3.2) so that we take their union. Their union is a graph, but taking its modulo leads to a generalized Cayley graph $F(X)$.

We now formalize this idea.
First, we must make sure that a local rule is an object that adopts the same naming conventions for vertices as those of the basic graph operations of Subsection 2.3.2.

Definition 23 (Dynamics non-modulo). *A function f from $\mathcal{X}_{\Sigma,\Delta,\pi}$ to $\mathcal{G}_{\Sigma,\Delta,\pi}$ is said to be a* dynamics *if and only if for all X the vertices of $f(X)$ are disjoint subsets of $V(X).S$, and $\varepsilon \in f(X)$.*

Intuitively, the integer $z \in S$ stands for the 'successor number z'. Hence the vertices designated by $\{1\}, \{2\} \cdots$ are successors of the vertex ε, whereas $\{\varepsilon\}$ is its 'continuation', i.e. its direct descendant. The vertices designated by $\{ab.1\}, \{ab.2\} \cdots$ are successors of its neighbour $ab \in X^r$. A vertex named $\{1, ab.3\}$ is understood to be both the first successor of vertex ε and the third successor of vertex ab. Recall also that ε, just like ab, are not just words but entire equivalence classes of these words, i.e. elements of $V(X)$.

Next, we disallow local rules that would suddenly produce an infinite graph.

Definition 24 (Boundedness non-modulo). *A function f from $\mathcal{X}^r_{\Sigma,\Delta,\pi}$ to $\mathcal{G}_{\Sigma,\Delta,\pi}$ is said to be* bounded *if and only if for all X, the graph $f(X)$ is finite.*

Finally, we make sure that the different pieces of graphs that are produced by the local rule are consistent with one another.

Definition 25 (Local rule). *A function f from $\mathcal{X}_{\Sigma,\Delta,\pi}$ to $\mathcal{G}_{\Sigma,\Delta,\pi}$ is a* local rule *if and only if it is a bounded dynamics and*

- *For any disk X^{r+1} and any $u \in X^0$ we have that $f(X^r)$ and $u.f(X^r_u)$ are non-trivially consistent.*
- *For any disk X^{3r+2} and any $u \in X^{2r+1}$ we have that $f(X^r)$ and $u.f(X^r_u)$ are consistent.*

It is clear that we do not need to formulate any consistency condition beyond $u \in X^{2r+1}$, because $f(X^r)$ and $u.f(X^r_u)$ then become trivially consistent, as they share nothing in common, see Figure 3.3. The only subtlety in Definition 25 is to impose that within $u \in X^0$, the produced pieces of graphs $f(X^r)$ and $u.f(X^r_u)$ be non-trivially consistent, i.e. consistent and overlapping (see Figure 3.3). The point here is to enforce the connectedness of the union of the pieces of graphs via a local, syntactic restriction. To illus-

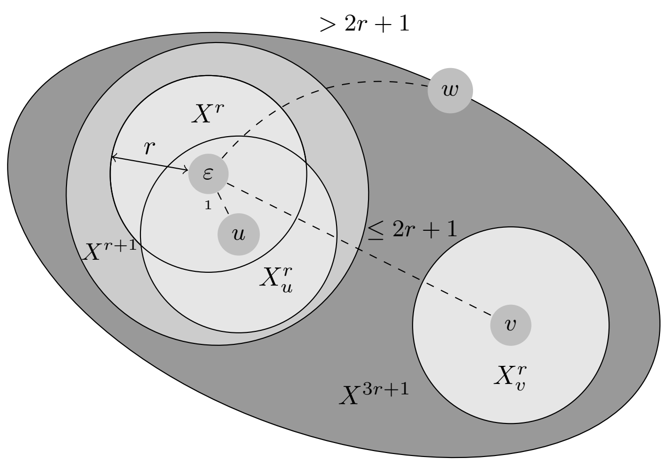

Figure 3.3: *The consistency conditions for a local rule.* The drawing represents disks of a generalized Cayley graph X upon which a local rule f of radius r will be applied. $f(X^r)$ and $u.f(X^r_u)$ have to be non-trivially consistent since ε and u are at distance 1. $f(X^r)$ and $v.f(X^r_v)$ have to be consistent but their intersection is allowed to be empty. $f(X^r)$ and $w.f(X^r_w)$ will be trivially consistent as they are to far to interact in one time step. The disk X^{r+1} is large enough to check all the non-trivial consistency conditions, as it contains first neighbours and their r-disks. The disk X^{3r+1} is enough to check all the consistency conditions, as it contains all the $2r+1$ neighbours and their r-disks.

trate the concept of local rule, we will now describe a local rule implementing the inflating grid dynamics. The local rule is of radius zero: it "sees" the neighbour vertices and nothing more. In the standard case the local rule is applied on a vertex surrounded by four neighbours. It then generates a graph of twelve vertices, each with particular names (see Figure 3.4). In particular cases, when less than four neighbours are present, the rule generates a graph of 10, 8, 6 or 4 vertices, each with particular names (see Figure 3.5). The local rule is not exhaustively described here, since there exists 625 different neighbourhoods of radius 0. In any case, all generated vertex names are carefully chosen, so that when taking the union of all the generated subgraphs, the name collisions lead to the desired identification of vertices (see Figure 3.6).

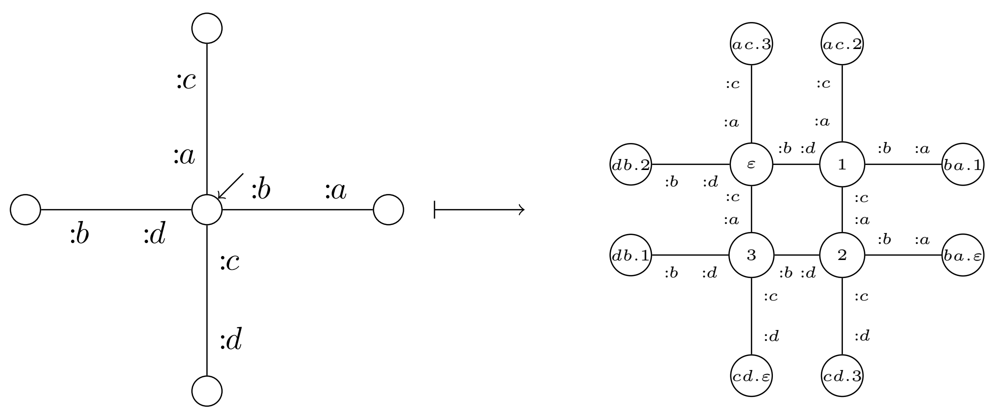

Figure 3.4: *Standard case of the inflating grid local rule.* The left-hand-side of the rule is a generalized Cayley graph of form X^0_u (a disk of radius 0). The right-hand-side is a graph whose vertex names are subsets of $V(X^0_u).S$. Here they are just singletons, curly brackets are dropped: e.g. we wrote $ac.3$ for $\{ac.3\}$, which should be understood as "the third successor of my neighbour on edge ac".

Definition 26 (Localizable function). *A function F from $\mathcal{X}_{\Sigma,\Delta,\pi}$ to $\mathcal{X}_{\Sigma,\Delta,\pi}$ is said to be* localizable *if and only if there exists a radius r and a local rule f from $\mathcal{X}^r_{\Sigma,\Delta,\pi}$ to $\mathcal{G}_{\Sigma,\Delta,\pi}$ such that for all X, $F(X)$ is given by the equivalence*

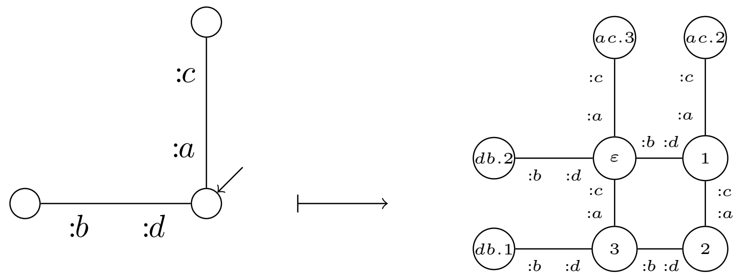

Figure 3.5: *A particular case of the inflating grid local rule.*

class modulo isomorphism, of the pointed graph

$$\bigcup_{u \in X} u.f(X_u^r)$$

with ε taken as the pointer.

3.3 Equivalence theorem

The following theorem shows that the constructive definition (*localizable* functions) is in fact equivalent to the mathematical, axiomatic definition (*causal* dynamics).

Theorem 1 (Causal is equivalent to localizable). *Let F be a function from $\mathcal{X}_{\Sigma,\Delta,\pi}$ to $\mathcal{X}_{\Sigma,\Delta,\pi}$. The function F is localizable if and only if there exists $R_\bullet$ such that $(F, R_\bullet)$ is a causal dynamics.*

Proof. [**Loc.⇒Caus.**] Let $F : \mathcal{X}_{\Sigma,\Delta,\pi} \to \mathcal{X}_{\Sigma,\Delta,\pi}$ be a localizable dynamics with local rule f from $\mathcal{X}^r_{\Sigma,\Delta,\pi}$ to $\mathcal{G}_{\Sigma,\Delta,\pi}$. $F(X)$ is the equivalence class, with ε taken as the pointer vertex, of the graph $H(X) = \bigcup u.f(X_u^r)$.

[**Dynamics**] Using the dynamicity of the local rule f, for all X^r we have $\varepsilon \in f(X^r)$. Therefore, for all $u \in X$, we have $u \in u.f(X^r)$ and thus $u \in H(X)$. Let R be an isomorphism such that $G(F(X)) = RH(X)$. Let

$u \in V(X)$, we define $R_X(u)$ to be $R(u')$, where u' is the vertex of $H(X)$ that contains u in its name. Notice that $\widetilde{(H(X), u)} = \widetilde{(R_X H(X), R_X(u))} = \widetilde{(G(F(X)), R_X(u))} = F(X)_{R_X(u)}$.

[**Translation-invariance**] Take $u \in X$. We have $H(X_u) = \bigcup v.f(X^r_{u.v})$. This is equal to $H(X_u) = \overline{u}.\bigcup u.v.f(X^r_{u.v})$, which in turn is equal to $\overline{u}.H(X)$. Next, we have that $F(X_u) = \widetilde{(H(X_u), \varepsilon)} = \widetilde{(\overline{u}.H(X), \overline{u}.u)} = \widetilde{(H(X), u)} = F(X)_{R_X(u)}$. It follows that $F(X_u) = F(X)_{R_X(u)}$, and so $G(F(X_u)) = \overline{R_X(u)}.G(F(X))$. We have therefore

$$G(F(X)) = R_X(u).G(F(X_u)) = R_X(u).R_{X_u}H(X_u) = R_X(u).R_{X_u}\overline{u}.H(X).$$

But since the relation $G(F(X)) = RH(X)$ defines R_X, we have proven that for all $u \in X$, $R_X = (R_X(u).R_{X_u}\overline{u}.)$. It follows that, for all $u.v \in X$, $R_X(u.v) = R_X(u).R_{X_u}(v)$.

[**Boundedness**] for all X, for all $w' \in F(X)$, consider $w \in H(X)$ such that $w' = R(w)$ when $G(F(X)) = RH(X)$, and $u \in X$ such that $w \in u.f(X^r_u)$. Since $\varepsilon \in f(X^r_u)$, we have $u \in u.f(X^r_u)$. Since f is bounded, w lies at most at a certain distance b of u in $H(X)$. Since $G(F(X) = RH(X)$, w' lies at most at a certain distance b of $u' = R(u) = R_X(u)$ in $F(X)$.

[**Continuity**] The following is illustrated in Figure 3.7. Let $m \in \mathbb{N}$. We must show that there exists an integer n such that $F(X)^m = \tilde{H}(X)^m_\varepsilon$ is determined by X^n.

Consider a sequence $v_0 = \varepsilon, v_1, \cdots, v_{m+1}$ of vertices of $H(X)$ such that for all $i \in \{0, \cdots, m\}$ there exists $e_i = (v_i : a_i, v_{i+1} : b_{i+1})$ in $E(H(X))$. For such an e_i to exist, and according to Definitions 15 and 16, it must appear in some $u_i.f(X^r_{u_i})$. Moreover if $\delta(e_i)$ is defined, it must be defined in some $u_i.f(X^r_{u_i})$. Consider $u_0, u_1, \cdots, u_m$ a sequence of vertices of X such that this is the case. Also, since v_{i+1} is a subset of $V(X).S$, there exists $w_i \in X, z_i \in S$ such that $w_i.z_i \in v_i$. Again consider $w_0 = \varepsilon, w_1, \cdots, w_{m+1}$ a sequence of vertices of X such that this is the case.

Since e_i is in $u_i.f(X_{u_i})$, it follows that v_i and v_{i+1} are in $u_i.f(X_{u_i})$. This entails that v_i and v_{i+1} are subsets of $u_i.V(X_{u_i}).S$, thus in particular $w_i, w_{i+1} \in u_i.V(X_{u_i})$. Therefore we have both $w_{i+1} \in u_i.X_{u_i}$ and $w_{i+1} \in u_{i+1}.X_{u_{i+1}}$. As a consequence u_i and u_{i+1} lie at distance $2(r+1)$ in X, and it follows that $\bigcup_{i=0\cdots m} u_i.X^r_{u_i} \subseteq X^{2(m+1)(r+1)-1}$. Hence $X^{2(m+1)(r+1)-1}$ determines

$E(H(X)^m_\varepsilon)$ and their internal states.
For $\sigma(v_i)$ to be defined, there must exists $x_i \in X$ such that $\sigma(v_i)$ is defined in $x_i.f(X^r_{x_i})$. Consider $x_0, x_1, \cdots x_m$ a sequence of vertices of X such that this is the case. But since $v_i \in x_i.f(X^r_{x_i})$, we must have that $w_i \in x_i.X^r_{x_i}$. Thus x_{j+1} lies at distance at most $r+1$ of $u_j.X^r_{u_j}$. Hence x_j lies at distance at most $r+1$ of $\bigcup_{i=0}^{m-1} u_i.X^r_{u_i} \subseteq X^{2m(r+1)-1}$. Hence $x_j \in X^{2m(r+1)+r}$, and thus $\bigcup_{i=0\cdots m} x_i.X^r_{x_i} \subseteq X^{2m(r+1)+2r+1}$. Hence $X^{2(m+1)(r+1)-1}$ determines the internal states of $H(X)^m_\varepsilon$.
Summarizing, X^n, with $n = 2(m+1)(r+1)-1$ determines $F(X)^m = \tilde{H}(X)^m_\varepsilon$. Consider some $v'' \in R^m_X$. This means that $v'' \in (RH(X))^m_\varepsilon$ and $v'' = R(v')$ for some $v' \in H(X)$ that contains $v \in X$ in its name. Hence $v' \in H(X)^m_\varepsilon$, where we used $R(\varepsilon) = \varepsilon$. Since this is determined by X^n, we have $v \in X^n$. Hence dom $R^m_X \subseteq X^n$. Moreover, consider X' such that $X'^r = X^r$. Therefore $v \in X'^r$, $H(X)^m_\varepsilon$ and $H(X')^m_\varepsilon$ are isomorphic, and this isomorphism sends v' to the w' of $H(X')^m_\varepsilon$ whose name contains v. Therefore $F(X)^m$ and $F(X')^m$ are equal, and the same paths designate $R^m_X(v)$ and $R^m_{X'}(v)$, which are thus equal.

[**Caus.⇒Loc.**] Let $(F, R_\bullet)$ be a causal dynamics. Let b_0 and b_1 be respectively the bounds given by Definition 21 and Lemma 2, and $b = \max(b_0 + 1, b_1)$. Let $m = 3b + 2$. Let r be the radius such that for all X, X', $X^r = X'^r$ implies $F(X)^m = F(X')^m$ and $R^m_X = R^m_{X'}$, from Definition 22 and Heine's Theorem. We will construct f from $\mathcal{X}^r$ to $\mathcal{G}_{\Sigma,\Delta,\pi}$ so that for all X^r, the graph $f(X^r)$ is a well-chosen member of the equivalence class $F(X^r)^b$. Hence we must instantiate $F(X^r)^b$ via a suitable, local naming of its vertices. We use the isomorphism S_{X^r} of Lemma 3 for this purpose, i.e. $f(X^r) = S_{X^r}G(F(X^r)^b)$.

[**Dynamics**] For all X^r, $f(X^r)$ has vertices that are subsets of $V(X^r).S$, by definition. These sets are disjoint, by Lemma 3 (i) applied to pairs of vertices of $F(X^r)^b$. Moreover $\varepsilon \in f(X^r)$, since $\varepsilon \in F(X^r)^b$ and $S_{X^r}(\varepsilon) = \varepsilon$ by Lemma 3 (ii).

[**Boundedness**] For all X^r, the graph $f(X^r)$ is finite, by construction.

[**Consistency**] In order to show the consistency of f, we will show that for all X, $u \in X$, we have that $u.f(X^r_u)$ is the subgraph $H(X)^b_u$ of $H(X)$, where

$H(X)$ is a well-chosen member of the equivalence class $F(X)$. Hence we must instantiate $F(X)$ via a suitable naming of its vertices. We use the isomorphism S_X of Lemma 3 for this purpose, i.e. $H(X) = S_X G(F(X))$.
Start from $u.f(X^r_u) = u.S_{X^r_u} G(F(X^r_u)^b)$, which is equal to $u.S_{X_u} G(F(X_u)^b)$, by Lemma 3 (iii) and using the fact that $F(Y)^b = F(Y^r)^b$. This, in turn, is equal to $u.\big(S_{X_u} G(F(X_u))\big)^b$, using the natural inclusion of $F(Y)^b$ into $F(Y)$. This, in turn, is equal to $u.\big(S_{X_u} \overline{R_X(u)}.G(F(X))\big)^b$, by shift-invariance, which is equal to $u.\big(\overline{u}.S_X G(F(X))\big)^b$, by Lemma 3 (iv). This, finally, is $\big(S_X G(F(X))\big)^b_u = H(X)^b_u$, since it is true that for any graph G and any isomorphism T, $TG^b_u = (TG)^b_{T(u)}$ and thus $G^b_u = T^{-1}(TG)^b_{T(u)}$.
Summarizing, $u.f(X^r_u) = H(X)^b_u$. Moreover if $u \in X^0$, then notice that $u \in f(X^r)$ and $u \in u.f(X^r_u)$, and hence they are non-trivially consistent.
Since f is consistent, and $f(X^r)$ is a representative of $F(X^r)^b$, it remains only to remark that $F(X) = \bigcup u.F(X^r_u)^b$, which is true because b was chosen to be strictly larger than the one given by Definition 21, insuring that all the vertices and edges of $F(X^r)$ are covered, along with their labels. □

In the proof of Theorem 4, the renaming S_X takes a generalized Cayley graph $F(X)$ into a mere graph $H(X)$. It does so by providing names for the vertices of $F(X)$, that are subsets of $V(X).S$. The idea is that w' in $F(X)$ gets named $S_X(w')$, which is the set of those $u.z$, such that $u' = R_X(u)$ is close to w', and z is an integer encoding the remaining path between u' and w'. Lemma 3 formalizes this idea as well as some useful, technical although expected properties.

Lemma 3 (Local renaming properties). *Let $(F, R_\bullet)$ be a causal dynamics. Let b be the maximum of the bounds from Definition 21 and Lemma 2. Let $m = 3b+2$. Let r be the radius such that for all X, X', $X^r = X'^r$ implies $F(X)^m = F(X')^m$ and $R^m_X = R^m_{X'}$, from Definition 22 and Heine's Theorem. Let z be an injection from $V(\mathcal{X}^b_\pi)\backslash\varepsilon$, as in Definition 10, to $\mathbb{N}$. Let $z(\varepsilon)$ be the empty word. Let Y be a generalized Cayley graph. Consider S_Y such that for all $w' \in F(Y)$ we have*

$$S_Y(w') = \{u.z(v') \,|\, u'.v' = w' \wedge u \in Y \wedge u' = R_Y(u) \wedge v' \in F(Y)^b_{u'}\}.$$

We have:

(i) $\forall w'_1, w'_2 \in F(Y),\ S_Y(w'_1) \cap S_Y(w'_2) \neq \emptyset \Rightarrow S_Y(w'_1) = S_Y(w'_2)$.

(ii) $\varepsilon \in S_Y(\varepsilon)$.

(iii) $\forall w' \in F(X_u)^b$, $u.S_{X_u^r}(w') = u.S_{X_u}(w')$.

(iv) $\forall v' \in F(X_u)$, $S_X(R_X(u).v') = u.S_{X_u}(v')$.

Proof. [**(i)**] Consider w'_1, w'_2 such that $S_Y(w'_1)$ and $S_Y(w'_2)$ have a common element $u.z(v')$. This entails that $w'_1 = u'.v' = w'_2$ is the same vertex in $F(Y)$, and thus that $S_Y(w'_1) = S_Y(w'_2)$.
[**(ii)**] Since $z(\varepsilon) = \varepsilon$, $\varepsilon.\varepsilon = \varepsilon$, $\varepsilon = R_Y(\varepsilon)$ and $\varepsilon \in F(Y)^b$.
[**(iii)**] Consider the $u = \varepsilon$ case. Let w' be a vertex of $F(X)^b$, and $u' \in F(X)$ a vertex such that $u'.v' = w'$, with $|v'| \leq b+1$. We necessarily have that $u' \in F(X)^{2b+1}$. Moreover, since $F(X)^{3b+2} = F(X^r)^{3b+2}$, we have $F(X)^b_{u'} = F(X^r)^b_{u'}$. Also, using $R^m_X = R^m_{X^r}$, we have that

$$u' = R_X(u) \Leftrightarrow u' = R^m_X(u) \Leftrightarrow u' = R^m_{X^r}(u) \Leftrightarrow u' = R_{X^r}(u).$$

where the middle equivalence uses the natural inclusion of X^r into X. As a consequence the two sets:

$$\begin{aligned} S_X(w') &= \{u.z(v') \,|\, u'.v' = w' \wedge u \in X \wedge u' = R_X(u) \wedge v' \in F(X)^b_{u'}\} \\ S_{X^r}(w') &= \{u.z(v') \,|\, u'.v' = w' \wedge u \in X^r \wedge u' = R_{X^r}(u) \wedge v' \in F(X^r)^b_{u'}\} \end{aligned}$$

are equal, up to the natural inclusion of X^r into X. The same holds for S_{X_u} and $S_{X_u^r}$. Then, since the shift operation $(u.)$ is from $V(X^n)$ to $V(X)$, a full equality holds between $u.S_{X_u}$ and $u.S_{X_u^r}$.
[*(iv)*] Consider some $u'.v'.w' \in F(X)$ with $u' = R_X(u)$, $v' = R_{X_u}(v)$ and $w' \in F(X_{u.v})^b$.

$$\begin{aligned} u.S_{X_u}(v'.w') &= u.\{x.z(y') \,|\, v'.w' = x'.y' \wedge x \in X_u \\ &\qquad \wedge x' = R_{X_u}(x) \wedge y' \in F(Y)'^b_u\} \\ &= \{u.x.z(y') \,|\, u'.v'.w' = u'.x'.y' \wedge u.x \in X \\ &\qquad \wedge u'.x' = R_X(u.x) \wedge y' \in F(Y)'^b_u\} \\ &= S_X(u'.v'.w') \\ &= S_X(R_X(u).v'.w') \end{aligned}$$

$\square$

Our causal dynamics over generalized Cayley graphs is a candidate model of computation accounting for space, but without this space being fixed. As a candidate model of computation, we must check that it is computable. The following shows that we can decide whether a syntactic object is a valid instance of the model.

Proposition 1 (Decidability of consistency). *Given a dynamics f from $\mathcal{X}^r_{\Sigma,\Delta,\pi}$ to $\mathcal{G}_{\Sigma,\Delta,\pi}$, it is decidable whether f is a local rule.*

Proof. First of all notice that there is a finite number of disks X^b of radius b, with labels in finite sets Δ and Σ. The following informal procedure verifies that f is a local rule:

- For each X^r check that $\varepsilon \in f(X^r)$.
- For each X^{r+1} check that for all $u \in X^0$, $f(X^r)$ and $u.f(X^r_u)$ are non-trivially consistent.
- For each X^{3r+2} check that for all $u \in X^{2r+1}$, $f(X^r)$ and $u.f(X^r_u)$ are non-trivially consistent.

□

Finally, we prove that if the initial state is finite, its evolution can be computed.

Proposition 2 (Computability of causal functions). *Given a local rule f and a finite generalized Cayley graph X, then $F(X)$ is computable, with F the causal graph dynamics induced by f.*

Proof. Since f is a local rule, the images of disks of radius r included in X are all finite, and consistent with one another. Moreover the finite union of finite, consistent graphs, is computable. □

3.4 Properties

Composability. We have characterized causal dynamics as the continuous, shift-invariant, bounded functions over generalized Cayley graphs. An important question is whether this notion is general enough. A good indicator of this robustness is its stability under composition.

Definition 27 (Composition). *Consider two dynamics* $(F, R_\bullet)$ *and* $(G, S_\bullet)$*. Their composition* $(G, S_\bullet) \circ (F, R_\bullet)$ *is* $(G \circ F, T_\bullet)$ *where* $T_X = S_{F(X)} \circ R_X$*, i.e.* $T_X(v) = S_{F(X)}(R_X(v))$*.*

Indeed, stability under composition holds for classical and reversible cellular automata, but has failed to be obtained for the early definitions of probabilistic cellular automata and quantum cellular automata (see [AFNT11] and [DS96, SW04, ANW08] for a discussion).

Theorem 2 (Composability). *[AD12] Consider causal dynamics* $(F, R_\bullet)$ *and* $(G, S_\bullet)$*, both over* $\mathcal{X}_{\Sigma,\Delta,\pi}$*. Then their composition is also a causal dynamics.*

Proof. [**Continuous**] In the $F', G' : \mathcal{X}_{\Sigma',\Delta,\pi} \to \mathcal{X}_{\Sigma',\Delta,\pi}$ formalism, it suffices to state that the composition of two continuous functions is continuous. Without this formalism this decomposes into:

- $(G \circ F)$ is continuous because it is the composition of two continuous functions.
- Consider $T_\bullet = S_{F(\bullet)} \circ R_\bullet$. For all X, for all m, there exists n such that for all X', $X'^n = X^n$ implies $T^m_{X'} = T^m_X$. Indeed:

Fix some X and m. Since $(G, S_\bullet)$ is a causal dynamics, there exists a radius n' such that for all X', $F(X')^{n'} = D' = F(X)^{n'}$ implies $S^m_{F(X')} = S^m_{D'} = S^m_{F(X)}$. Fix this n'. Since $(F, R_\bullet)$ is a causal dynamics, there exists a radius n such that for all X', $X^n = D = X'^n$ implies $F(X)^{n'} = F(X')^{n'}$ and $R^{n'}_X = R^{n'}_D = R^{n'}_{X'}$. Now, for this radius n, $T^m_{X'} = S^m_{F(X')} \circ R^{n'}_{X'} = S^m_{D'} \circ R^{n'}_{X'} = S^m_{D'} \circ R^{n'}_D$, which, by the symmetrical is equal to T^m_X.
[**Shift-invariant**] We have $G(F(X_u)) = G(F(X)_{R_X(u)}) = G(F(X))_{S_{F(X)}(R_X(u))}$,

$$\begin{aligned}
T_X(u.v) = &\quad S_{F(X)}(R_X(u.v)) \\
= &\quad S_{F(X)}(R_X(u).R_{X_u}(v))) \\
= &\quad S_{F(X)}(R_X(u)).S_{F(X)_{R_X(u)}}(R_{X_u}(v)) \\
= &\quad T_X(u).S_{F(X_u)}(R_{X_u}(v)) \\
= &\quad T_X(u).T_{X_u}(v)
\end{aligned}$$

[**Bounded**] Since $(G, S_\bullet)$ is a causal dynamics, there exists a bound b'' such that for all X, for all $w'' \in G(F(X))$, there exists $x'' = S_{F(X)}(x')$ and $v'' \in$

$G(F(X))^{b''}_{x''}$ such that $w'' = x''.v''$. Since $(F, R_\bullet)$ is a causal dynamics, there exists a bound b' such that there exists $u' = S_{F(X)}(u)$ and $v' \in F(X)^{b''}_{u'}$ such that $x' = u'.v'$. Let $u'' = S_{F(X)}(u') = S_{F(X)}(R_X(u)) = T_X(u)$. Now, according to Lemma 2 applied to $(G, S_\bullet)$ and points u' and x', there exists a bound c such that there exists $t'' \in G(F(X))^{c.(b'+1)}_{u''}$ and $x'' = u''.t''$. Let $b = c.(b'+1)+b''$, we now have that for $u'' = S_{F(X)}(u') = S_{F(X)}(R_X(u)) = T_X(u)$ there exists $v''.t'' \in G(F(X))^b_{u''}$ such that $w'' = u''.t''.v''$. $\square$

The above proof was done via the axiomatic characterization of causal dynamics, as this work enjoys a more straightforward formalization than [AD12]. In [AD12] the same result is proven via the constructive approach to causal graph dynamics (localizability), which has the advantage of extra information about the composed function. It establishes the following. Consider F a causal dynamics induced by the local rule f of radius r (i.e. diameter $d = 2r + 1$). Consider G a causal graph dynamics induced by the local rule g of radius s (i.e. diameter $e = 2s + 1$). Then $G \circ F$ is a causal graph dynamics induced by the local rule g of radius $t = 2rs + r + s$ (i.e. diameter $f = de$) which maps X^t to

$$\bigcup_{v \in X'} v.g(X'^s_v) \quad \text{with} \quad X' = \bigcup_{u \in X^t} u.f(X^r_u).$$

The same result, with the transposed proof, still holds.

3.5 Summary of results

We gave two definitions of causal graph dynamics as cellular automata over generalized Cayley graphs. While the first definition, causal dynamics, is axiomatic and correspond to graph transformations having three properties (uniform continuity, shift-invariance and boundedness), the second definition is more constructive and relies on a notion of local rule applied simultaneously on every vertex in the graph. We then prove the equivalence of these two definitions, In addition, we proved that the application of a causal graph dynamics on a finite graph is a computable process, and that the composition of two causal graph dynamics is also a causal graph dynamics.

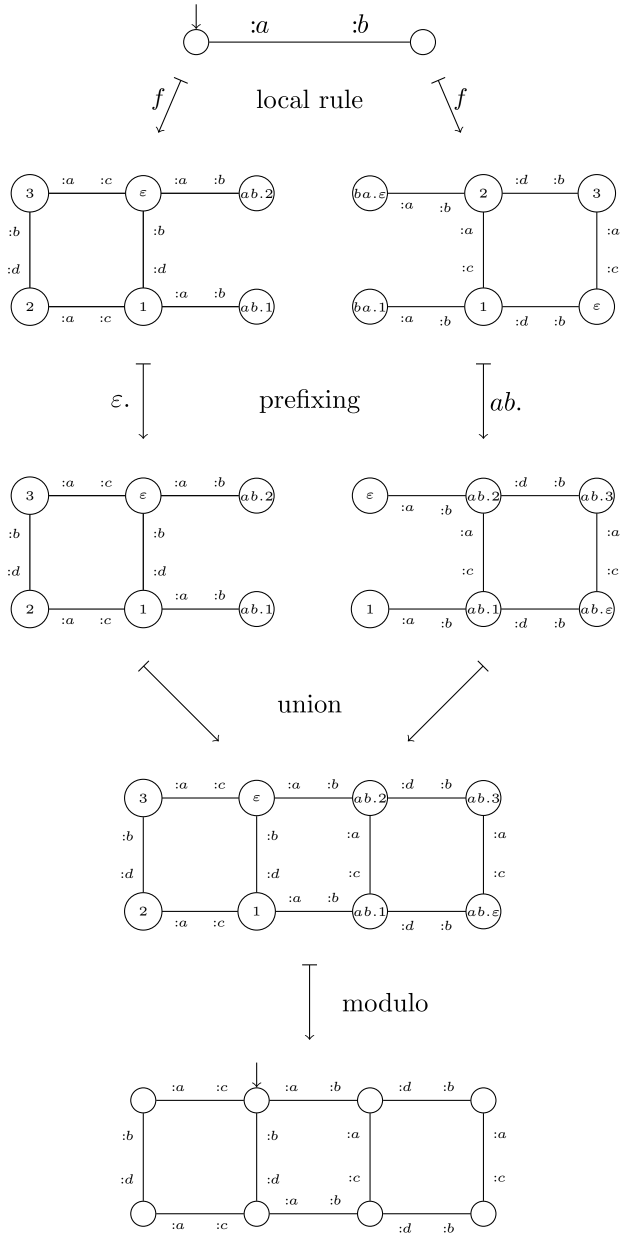

Figure 3.6: *Local rule implementation of the inflating grid dynamics.* First the local rule is applied on the neighbourhood of every vertex of the input graph. The resulting graphs are prefixed (see definition 12) by the vertex they are issued of. Third a union of graphs is performed to obtain the output graph. Lastly, the corresponding pointed graph modulo is returned.

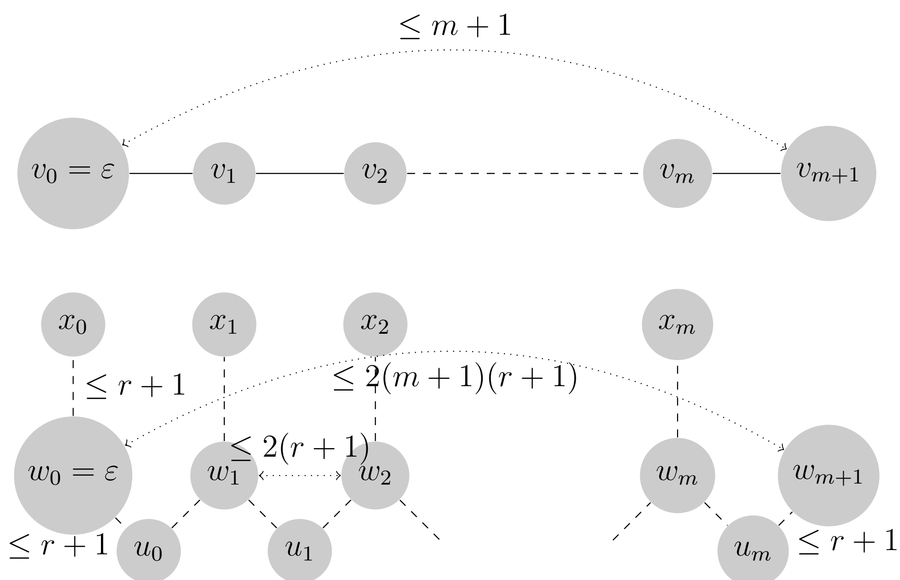

Figure 3.7: *Proof of continuity.*

Chapter 4

Reversible causal graph dynamics

Motivations. Cellular automata find their origins in physics, where they are commonly used as toy models for waves or particles interaction. In fact they are some of the simplest discrete models of physics implementing two of its symmetries: causality and homogeneity (see introduction). In this context, it seems natural to consider a further physics-like symmetry, namely reversibility. This symmetry has been studied in depth, and led to some beautiful results on the structure of reversible parallel computation, such as the block decomposition of reversible cellular automata. In the case of causal graph dynamics, reversibility brings novel questions. For instance, now that the configuration (the graph) can be extended by adding or deleting vertices, does bijectivity brings any constraints on the structure of the transformation? More generally, these questions are linked to a more fundamental one, coming from theoretical physics: is it possible to reconciliate reversible small scale physics (quantum mechanics, micro-mechanical), with the time-varying topology of large scale physics (relativity)? In that precise scope, defining a proper block decomposition for reversible causal graph dynamics seems to be a key step toward the design of a quantum version of causal graph dynamics.

Invertibility versus reversibility. As previously introduced, cellular automata can be constructively defined as homogeneous mappings of a local rule over a configuration, but, thanks to Curtis-Hedlund-Lyndon theorem (C.-H.-L.), they can also be defined as transformations over the configuration

space verifying two properties: continuity with respect to Cantor's metric, and commutation with the shift operator. That latter definition of the model comes in handy when studying reversibility. The first natural question we can ask ourselves is the following: is the bijectivity of the global function sufficient to ensure that its inverse is still a cellular automaton? The answer is yes, and the proof relies on a simple topological result: If F is a continuous function over a compact space X and if F is a bijection, then F^{-1} is continuous. Using C.-H.-L. theorem together with this result, we have that the inverse of the global function of a reversible cellular automata is continuous. The shift-commutation of the inverse of the global function is easy to obtain from the shift-commutation of the global function. Hence, any cellular automaton whose global function is bijective is reversible.
The situation is more complicated in the case of causal graph dynamics. While it is still true, using the same topological result, that the inverse of a causal graph dynamics is continuous, its shift-invariance is not obvious. Section 4.1 studies in details this shift-invariance property in order to prove two results:

- Every bijective causal graph dynamics preserves the number of vertices of all graphs larger than a certain bound,
- Every bijective causal graph dynamics admits an inverse causal graph dynamics.

Block decomposition. Even though C.-H.-L. theorem states that defining a CA using a local rule or describing a continuous shift-invariant transformation is equivalent, some antisymmetry between the two formalisms remain. Indeed, if we consider a reversible cellular automaton, and a local rule inducing it, nothing guarantees that this local rule will be itself a reversible function. In fact, as this function is of the form $f : \Sigma^{2r+1} \to \Sigma$, it is, in general, not the case. Hence, another natural question would be: is it possible to implement any reversible cellular automaton using only local, reversible mechanisms? In [Kar96, DL01], it is proved that reversible cellular automata admit a finite-depth, reversible circuit form, with gates acting only locally. The result carries through to Quantum CA [ANW10], whose proof technique inspired the construction described in [AN11].
In the scope of causal graph dynamics, the freedom of changing the shape of the underlying graph may seem to be a direct hurdle toward the generalization of the results of the above papers. Section 4.2 provides a definition of

local transformations in a graph and a notion of conjugate operation through a reversible causal graph dynamics in order to obtain a block decomposition of any reversible causal graph dynamics.

The curse of vertex-preservingness. Even though one result of section 4.1 generalizes a famous result of cellular automata theory to causal graph dynamics, the other result seems to put a consequent constraint upon the shape of reversible causal graph dynamics: reversible causal graph dynamics necessarily preserve the size of almost all the graphs. Section 4.3 describes a way to slightly generalize the model of causal graph dynamics in order to allow reversible dynamics to have a less constraint behaviour.

The content of this chapter is based on [AMP15], co-authored with Pablo Arrighi and Simon Perdrix.

4.1 Reversible causal graph dynamics

In this section we compare the two notions of invertibility and reversibility, defined as follows. An invertible causal graph dynamics, is a causal graph dynamics inducing a bijection over the set of generalized Cayley graphs:

Definition 28 (Invertible dynamics). *A causal graph dynamics* $(F, R_\bullet)$ *is said to be invertible if* F *is a bijection from* $\mathcal{X}_{\Sigma,\Delta,\pi}$ *to itself.*

While a causal graph dynamics $(F, R_\bullet)$ is said reversible if it is invertible *and* its inverse a causal graph dynamics itself:

Definition 29 (Reversible). *A causal graph dynamics* $(F, R_\bullet)$ *is* reversible *if there exists* $S_\bullet$ *such that* $(F^{-1}, S_\bullet)$ *a causal graph dynamics.*

Moving head. Figure 4.1 is an example of invertible causal graph dynamics. In this example, a vertex, representing the head of an automaton, is moving along a path graph, representing a tape. The path graph is built using $ab-$edges, while the head is attached using either a $cc-$edge if it is travelling forward along the $ab-$edges, or $dd-$edges if it is travelling backwards. The transformation can be completed into a bijection over the entire set of graphs with $\pi = \{a, b, c, d\}$. It then accounts for several heads, etc.

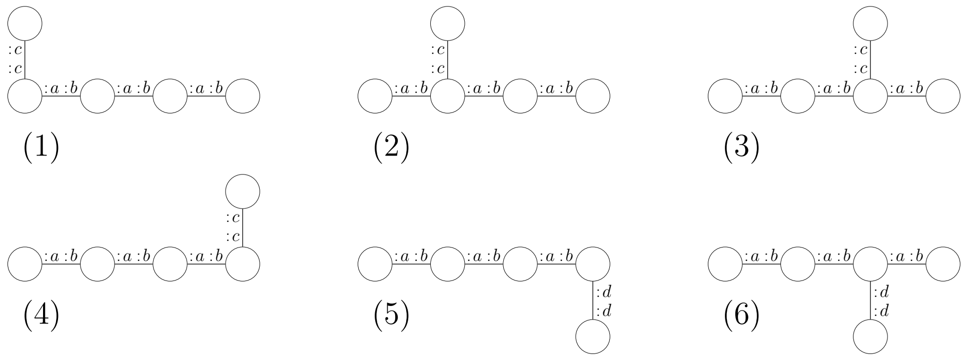

Figure 4.1: *Moving head dynamics.* In this example, a moving head is running along a “tape” formed by a linear graph of alternating *ab* edges. When reaching the end of the line, the head starts moving backwards and changes the ports on its attaching edge to *dd*. (1) to (6) represent 6 consecutive configurations.

4.1.1 Invertibility and almost-vertex-preservingness

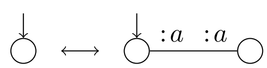

Figure 4.2: The *turtle dynamics* has the two above pointed graphs modulo to oscillate between one another. The two vertices of the right hand side are shift-equivalent, i.e. pointing the graph upon one or the other does not change the graph.

Recall that, in general, causal graph dynamics are allowed to transform the graph, not only by changing internal states and edges, but also by creating or deleting vertices. Since invertibility imposes information-conservation, one may wonder whether invertible causal graph dynamics are still allowed to create or delete vertices. They are, as shown by Figure 4.2. One notices, however, that the right hand side of this example features shift-equivalent vertices:

Definition 30 (Shift-equivalent vertices). *Let $X \in \mathcal{X}_{\Sigma,\Delta,\pi}$ and let $u, v \in V(X)$. We say that u and v are* shift-equivalent*, denoted $u \approx v$, if $X_u = X_v$. A graph is called* asymmetric *if it has only trivial (i.e. of size one) shift-equivalence classes.*

One can show that all the shift-equivalence classes of a pointed graph modulo have the same size. Intuitively, given two shift-equivalent vertices u, v and a third vertex w, since there is a path from u to w, moving from v along the same path leads to a vertex equivalent to w.

Lemma 4 (Shift-equivalence classes isometry). *Let $X \in \mathcal{X}_{\Sigma,\Delta,\pi}$ be a graph. If $C_1 \subseteq V(X)$ and $C_2 \subseteq V(X)$ are two shift-equivalence classes of X, then $|C_1| = |C_2|$.*

Proof. Consider two equivalent and distinct vertices u and v in X. Consider a path w. The vertices $u.w$ and $v.w$ are distinct and equivalent. More generally, if we have n equivalent distinct vertices $v_1, ..., v_n$, any vertex $u = v_1.w$ will be equivalent to $v_2.w, ..., v_n.w$ and distinct from all of them, hence the equivalence classes are all of the same size. □

Moreover, we can show that creation or deletion of vertices by invertible causal graph dynamics must respect the shift-symmetries of the graph.

Lemma 5 (Invertible causal graph dynamics preserves shift-equivalence classes). *Let $(F, R_\bullet)$ be a shift-invariant dynamics over $\mathcal{X}_{\Sigma,\Delta,\pi}$, such that F is a bijection. Then for any X and any $u, v \in X$, $u \approx v$ if and only if $R_X(u) \approx R_X(v)$.*

Proof. $u \approx v$ expresses $X_u = X_v$, which by bijectivity of F, is equivalent to $F(X_u) = F(X_v)$ and hence $F(X)_{R_X(u)} = F(X)_{R_X(v)}$. This, in turn, is expressed by $R_X(u) \approx R_X(v)$. □

Shift-symmetry is fragile however, and can be destroyed by adding a few vertices to a graph:

Definition 31 (Primal extension). *Given a finite graph $X \in \mathcal{X}_{\Sigma,\Delta,\pi}$ where $|\pi| > 1$ such that X has k shift-equivalence classes of size n with $k, n \neq 1$, we obtain its* primal extension *${}^\square X$ by:*

- *If X has a free port (i.e. one of its vertex u has a port $i \in \pi$ such that $u : i$ does not appear in any edge): connect $p - kn$ new vertices in a line to this free port, where p is the smallest prime number greater than $kn + 2$.*

- *If X has no free port: X has at least one cycle. Remove an edge from this cycle, and do the same construction as above.*

Lemma 6 (Properties of primal extensions). *Any primal extension* $^{\square}X$ *is asymmetric.*

Proof. As $^{\square}X$ has a prime number of vertices, by lemma 4, its has either one single equivalence class of maximal size or only trivial equivalence classes. As the primal extension adds at least two vertices and that these vertices have different degree (1 for the last vertex on the line, and 2 for its only neighbour), $^{\square}X$ contains at least two non equivalent vertices, hence the first result. □

Theorem 3 (Invertible implies almost-vertex-preserving). *Let* $(F, R_{\bullet})$ *be a causal graph dynamics over* $\mathcal{X}_{\Sigma,\Delta,\pi}$*, such that* F *is a bijection. Then there exists a bound* p*, such that for any graph* X*, if* $|X| > p$ *then* R_X *is bijective.*

Proof outline. We consider a finite graph X as large as we want, and assume that R_X is not a bijection. We the consider the two cases where R_X is not surjective and R_X is not injective. In the non surjective case, we asymmetrize $F(X)$ and consider its pre-image, thus exhibiting two "close" graphs having necessarily very different images, contradicting the continuity of $R_{\bullet}$. In the non injective case, we asymmetrize X, thus producing a similar situation and contradicting the continuity of $R_{\bullet}$.

In a second time we extend the result to infinite graph using the continuity of $(F, R_{\bullet})$.

Proof. When $|\pi| \leq 1$, $\mathcal{X}_{\Sigma,\Delta,\pi}$ is finite so the theorem is trivial. So we assume in the rest of the proof that $|\pi| > 1$.
[**Finite graphs**] First we prove the result for any finite graph. By contradiction, assume that there exists a sequence of finite graphs $(X(n))_{n\in\mathbb{N}}$ such that $|X(n)|$ diverges and such that for all n, $R_{X(n)}$ is not bijective. As this sequence is infinite, we have that one of the two following cases is verified for an infinite number of n:

- $R_{X(n)}$ is not surjective,
- $R_{X(n)}$ is not injective.

• [$R_{X(n)}$ **not surjective**]. There exists a vertex $v' \notin \operatorname{im} R_{X(n)}$. Without loss of generality, we can assume that $|v'| < b$ where b is the bound from the boundedness property of F. Consider the graph $Y(n) = F^{-1}(^{\square}F(X(n)))$. Using uniform continuity of F^{-1} and $R_{\bullet}$, and the fact that $|X(n)|$ is as large as we want, we have that there exists an index n and a radius r such that $Y(n)^r = X(n)^r$ and $R^b_{Y(n)^r} = R^b_{X(n)^r}$. As $F(Y(n))$ is asymmetric by construction, $v' \in \operatorname{im} R^b_{Y(n)^r}$ which contradicts $v' \notin \operatorname{im} R_{X(n)}$.

• [$R_{X(n)}$ **not injective**]. There exist two vertices $u, v \in X(n)$ such that $R_{X(n)}(u) = R_{X(n)}(v)$ and $u \neq v$. Without loss of generality, we can assume that $u = \varepsilon$ as F is shift-invariant. According to lemma 5, we have that $\varepsilon \approx v$. Moreover, using the uniform continuity of $R_\bullet$, we have that, as $R_{X(n)}(v) = R_{X(n)}(\varepsilon) = \varepsilon$, there exists a radius l, which does not depend on n, such that $|v| < l$. Let us consider the graph ${}^\square X(n)$. In this graph, ε and v are not shift-equivalent and thus, $R_{{}^\square X(n)}(\varepsilon) \neq R_{{}^\square X(n)}(v)$. By continuity of $R_\bullet$, we have that there exists a radius $r > l$ such that $R^0_{{}^\square X(n)^r} = R^0_{X(n)^r}$ for a large enough n. Hence $R^0_{{}^\square X(n)^r}(v) = R^0_{X(n)^r}(v) = \varepsilon$, which contradicts $R_{{}^\square X(n)}(\varepsilon) \neq R_{{}^\square X(n)}(v)$.

[**Infinite graphs**] We now show that the result on finite graphs can be extended to infinite graphs, proving that for any infinite graph R_X is bijective:

• [R_X **injective**]. By contradiction. Take X infinite such that there is $u \neq v$ and $R_X(u) = R_X(v)$. Without loss of generality we can take $u = \varepsilon$, i.e. $v \neq \varepsilon$ and $R_X(v) = \varepsilon$. By continuity of $R_\bullet$, there exists a radius r, which we can take larger than $|v|$ and p, such that $R_X = R_{X^r}$. Then $R_{X^r}(v) = R_X(v) = \varepsilon$, thus R_{X^r} is not injective in spite of X^r being finite and larger than p, leading to a contradiction.

• [R_X **surjective**]. By contradiction. Take X infinite such that there is v' in $F(X)$ and $v' \notin \operatorname{im} R_X$. By boundedness and shift-invariance, we can assume that there exists b such that $|v'| < b$. By continuity of $R_\bullet$, there exists a radius r, which we can take larger than p, such that the images of R_X and R_{X^r} coincide over the disk of radius b. Then, $v' \notin \operatorname{im} R_X$ implies $v' \notin \operatorname{im} R_{X^r}$, thus R_{X^r} is not surjective in spite of X^r being finite and larger than p, leading to a contradiction.

□

4.1.2 Invertibility vs. Reversibility

Theorem 3 shows that invertible causal graph dynamics are almost vertex-preserving. Notice that vertex-preservingness guarantees that the inverse of a shift-invariant dynamics is a shift-invariant dynamics.

Lemma 7 (Vertex-preserving invertible is shift-invariant invert-

ible). *If $(F, R_\bullet)$ is an invertible shift-invariant dynamics such that for all X, R_X is a bijection, then $(F^{-1}, S_\bullet)$ is a shift-invariant dynamics, with $S_Y = (R_{F^{-1}(Y)})^{-1}$.*

Proof. Consider Y and $u'.v' \in Y$. Take X and $u.v \in X$ such that $F(X) = Y$, $R_X(u) = u'$ and $R_X(u.v) = u'.v'$. We have:

$$\begin{aligned} F^{-1}(Y_{u'}) &= F^{-1}(F(X)_{R_X(u)}) \\ &= F^{-1}(F(X_u)) \\ &= X_{(R_X)^{-1}(u')} \\ &= F^{-1}(Y)_{S_Y(u')} \end{aligned}$$

Moreover, take $v \in X_u$ such that $R_X(u.v) = R_X(u).R_{X_u}(v) = u'.v'$. We have:

$$\begin{aligned} S_Y(u'.v') &= (R_X)^{-1}(R_X(u.v)) \\ &= u.v \\ &= (R_X)^{-1}(u').(R_{X_u})^{-1}(v') \\ &= S_Y(u').S_{Y_{u'}}(v') \end{aligned}$$

□

Theorem 4 (Invertible implies reversible). *If $(F, R_\bullet)$ is an invertible causal graph dynamics, then $(F, R_\bullet)$ is reversible.*

Proof. Continuity of F^{-1} is directly given by the continuity of F together with the compactness of $\mathcal{X}_{\Sigma,\Delta,\pi}$. Its boundedness derives either from the bijectivity of R_X for $|X| > p$ or from the finiteness of X when $|X| \leq p$.

We must construct $S_\bullet$. For $|F(X)| = |X| > p$, we know that R_X is bijective and we let $S_{F(X)} = R_X^{-1}$. For $|X| \leq p$, we proceed as follows.
We write $\tilde{u}$ for the shift-equivalence class of u. For all $v' \in F(X)$, we make the arbitrary choice $S_{F(X)}(\tilde{v'}) = v$, where v is such that $R_X(v) \approx v'$. For this X, we have enforced $\approx$-compatibility. In order to enforce shift-invariance, we must make consistent choices for $S_{F(X)_{u'}}$. This is obtained by demanding that $S_{F(X)_{u'}}(\widetilde{\overline{u'}.v'}) = \overline{u}.v$. Indeed, this accomplishes shift-invariance because $S_{F(X)_{u'}}(v') = S_{F(X)_{u'}}(\overline{u'}.u'.v') = \varepsilon.v' = v'$ implying the equality: $S_{F(X)}(u'.v') = u.v = S_{F(X)}(u').S_{F(X)_{u'}}(v')$. Moreover, $S_{F(X)_{u'}}$ is itself shift-invariant because: $S_{F(X)_{u'.v'}}(w') = S_{F(X)_{u'.v'}}(\overline{v'}.v'.w') = \varepsilon.w = w$ and $S_{F(X)_{u'}}(v') = v$ implying that $S_{F(X)_{u'}}(v'.w') = v.w = S_{F(X)_{u'}}(v').S_{F(X)_{u'.v'}}(w')$

, and $\approx$-compatible because $v' \approx w'$ implies $S_{F(X)_{u'}}(v') = S_{F(X)_{u'}}(w')$, and thus $S_{F(X)_{u'}}(v') \approx S_{F(X)_{u'}}(w')$.
Continuity of the constructed $S_\bullet$ is due to the continuity of $R_\bullet$ and the finiteness of p.
Shift-invariance of $(F^{-1}, S_\bullet)$ follows from $\approx$-compatibility of $S_\bullet$ and shift-invariance of $(F, R_\bullet)$, because $F^{-1}(F(X)'_u) = X_v$ where v is such that $R_X(v) \approx u'$, hence $F^{-1}(F(X)'_u) = X_{S_{F(X)}(u')}$. □

4.2 Block decomposition of reversible causal graph dynamics

In [AN11], a general construction is presented, representing any reversible cellular automata G over $\Sigma^{\mathbb{Z}}$ as a finite-depth circuit of reversible local operation over the set $\Sigma^{2^{\mathbb{Z}}}$. The idea of the construction is the following. The circuit is composed of local update. Each local update acts upon a cell and its neighbourhood. Updating cell i corresponds to (1) applying G to the first component of the configuration, (2) swapping the two component of cell i and (3) applying G^{-1} on the first component of the configuration. After every cell has been updated, the second component of the configuration contains the image of the initial configuration. More intuitively, this corresponds to applying the CA around a cell and saving the new state of this cell in its second component. Figure 4.3 gives an example of such a decomposition. In our case, the concept of the construction will be somehow simpler. As our space of configurations is less constrained, we can proceed as follow: Apply the reversible causal graph dynamics F, "freeze" the vertex u, apply the reverse dynamics F^{-1}. As F is defined over the set of all graphs, it will be defined over the initial graph where u has been removed.

4.2.1 Locality

Causal graph dynamics change the entire graph in one go. Local operations, on the other hand, act just in one bounded region of the graph, leaving the rest unchanged. We introduce the following locality definition:

Definition 32 (Local dynamics). *A dynamics $(L, S_\bullet)$ is r-*local *if it is continuous and bounded, and for any X and any $v \in L(X)$ with $|v| > r$, there exists $u \in X$ such that $L(X)^0_v = X^0_u$ and $\forall w \in X^0_u$, $S_X(u.w) = v.w$.*

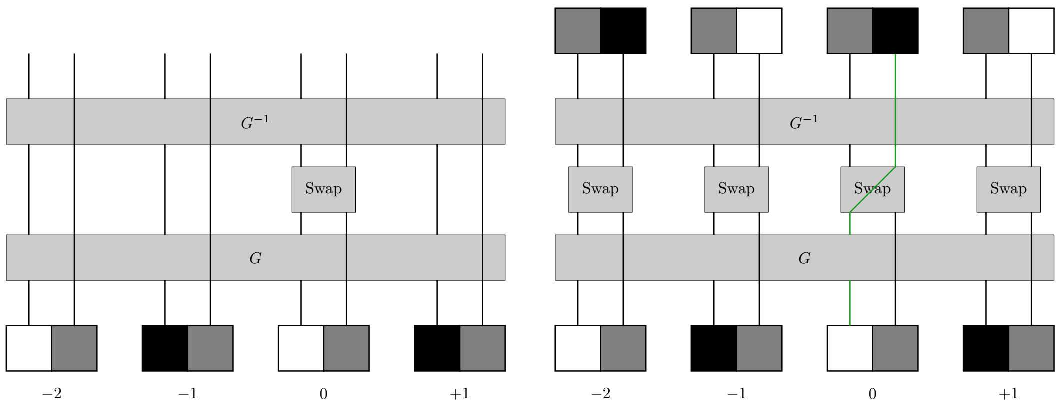

Figure 4.3: A simple block decomposition of a reversible cellular automaton. Each cell contains a product state in $\{0,1\}^2$. The initial configuration is encoded in the first component of the cells (left cells). Left hand side: Local reversible operation applied on cell 0. Right hand side: the complete circuit of local reversible operations.

Local operations may also be shifted to act over the region surrounding some vertex u. The details of the next definition become apparent with Figure 4.4.

Definition 33 (Shifted dynamics). *Consider a dynamics $(L, S_\bullet)$ and some $u \in \Pi^*$. We define L_u to be the map $X \mapsto (L(X_u))_{S_{X_u}(\overline{u})}$ if $u \in X$, and the identity otherwise. We define $S_{u,X}$ to be the map $v \mapsto \overline{S_{X_u}(\overline{u})}.S_{X_u}(\overline{u}.v)$ if $u \in X$, and the identity otherwise. We say that $(L_u, S_{u,\bullet})$ is $(L, S_\bullet)$* shifted *at u.*

A local dynamics acts around the pointer of the graph modulo. To act around another position u, one can shift the local dynamics at u. Moreover, we may wish to apply a series of local operations at several positions u_i i.e., a circuit. However, applying a local operation may change the graph and hence vertex names, hence some care must be taken when defining the successive application of local operations.

Definition 34 (Product). *Consider a local dynamics $(L, S_\bullet)$ and X a pointed graph modulo in its domain we define the product $\prod(L, S)$ as the*

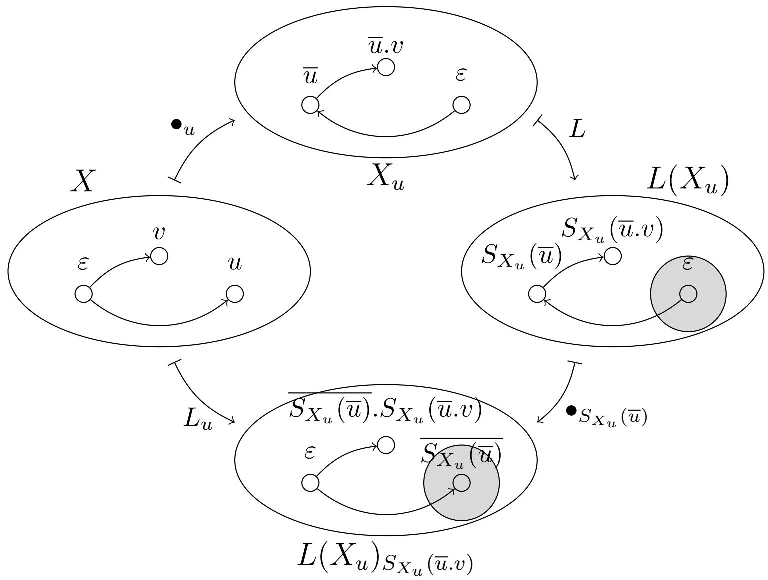

Figure 4.4: Shifted dynamics L_u. In the bottom graph $L_u(X)$, former vertex v has name $\overline{S_{X_u}(\overline{u})}.S_{X_u}(\overline{u}.v)$.

limit when r goes to infinity of $(L^r, S^r_\bullet)$:

$$L^r(X) = \prod_{i\in[1,\ldots,|V(X^r)|]} L_{u'_i}(X)$$

$$S^r_X = \prod_{i\in[1,\ldots,|V(X^r)|]} S_{u'_i,\prod_{k\in[1,\ldots,i-1]} L_{u'_k}(X)}$$

where $\{u_1, u_2, ...\} = V(X)$ such that $i < j \Rightarrow |u_i| \le |u_j|$, and $u'_1 = u_1$, $u'_2 = S_{u'_1,X}(u_2)$, $u'_3 = S_{u'_2,L_{u'_1}(X)}(u_3)$,...

Soundness of the definition. For infinite graphs, the image of a graph X through the application of $\prod(L,S)$ needs to be defined as the limit of the sequence of graphs $(L^r(X))$ obtained by applying L to every node of the disk X^r. As $\mathcal{X}_{\Sigma,\Delta,\pi}$ is compact, this sequence of graphs converges toward a limit graph X'. Moreover, for all radius r' there exists a radius r such that $X'^r = L^{r'}(X)$. Thus, X' corresponds to the graph X where the local dynamics $(L, S_\bullet)$ has been applied on every vertex.

4.2.2 Block representation

We can now start the construction of our local reversible updates. First, we show that conjugating a local operation with a reversible causal graph dynamics still yields a local operation.

Lemma 8 (Bounded inflation). *If $(L, S_\bullet)$ is r'-local, then for all s there exists s' such that for all X and $v \in X$, if $|v| \leq s$, then $|S_X(v)| \leq s'$.*

Proof. Suppose the contrary: $X(s')$ has some $|v(s')| \leq s$ such that $|S_X(v)| > s'$. Since $\mathcal{X}_{\Sigma,\Delta,\pi}$ is compact [AM12], $X(s')$ admits a subsequence which converges to some limit X, in the sense that $X(s'_k)^k = X^k$. For this particular X, for any s', there is some $|v(s')| \leq s$ such that $|S_X(v)| > s'$. This is because we can choose k so that $s'_k \geq s'$ and k superior to the radius needed to determine $L(X)^{s'_k} = L(X(s'_k))^{s'_k}$, so that $|S_X(v)| = |S_{X^k}(v)| = |S_{X(s_k)^k}(v)| = |S_{X(s_k)}(v)| > s'_k \geq s'$. Thus, there exists a point of $v \in X$ which has $|S_X(v)| > \infty$, which is a contradiction. □

Lemma 9 (Local at radius t). *If $(L, S_\bullet)$ is r'-local, for all t, for all $u' \in L(X)$ with $|u'| > r' + t + 1$, there exists $u \in X$ with $S_X(u) = u'$ such that we have:*

(i) $L(X)^t_{u'} = X^t_u$,

(ii) $\forall v \in X^t_u,\ S_X(u.v) = u'.v$.

Proof. Take such a u' and consider u such that $u' = S_X(u)$.
[*(i)*] Since $|u'| > r'+t+1$, we have that for all $v \in L(X)^t_{u'}$, $|u'.v| > r'$. Hence, by r'-locality of L, there exists $x \in X$ such that $S_X(x) = u'.v$ and such that $L(X)^0_{u'.v} = X^0_x$, i.e. the vertex v in $L(X)^t_{u'}$, in terms of its internal states and edges, is the same as the vertex x in X. Now, say there exists $|z| = 1$ such that $w = v.z \in L(X)^t_{u'}$, i.e. there is an edge between v and $v.z$ in $L(X)^t_{u'}$. Again since $|S_X(x)| > r'$, the r'-locality yields $u'.v.z = S_X(x).z = S_X(x.z)$, i.e. the edge between v and $v.z$ v in $L(X)^t_{u'}$ is the same as that between x and $x.z$ in X. Consider $v_1 \ldots v_k = v$ with $k \leq t$ and $|v_i| = 1$. A similar argument starting from u' and following these edges shows that x is at distance t of u in X, and thus $x.z$ is at distance $t+1$ of u in X. So the vertices x, $x.v$ and their edge do appear in X^t_u.
[*(ii)*] Again take $w \in X^t_u = L(X)^t_{u'}$. Consider $w_1 \ldots w_k = w$ with $k \leq t+1$ and $|w_i| = 1$. Since $|u'| > r' + t + 1 > r'$, the r'-locality applies

and yields $S_X(u.w_1) = S_X(u).w_1 = u'.w_1$. Similarly, since $|u'.w_1 \dots w_i| > r' + t + 1 - i > r'$, the r'-locality applies and yields $S_X(u.w_1 \dots w_i.w_{i+1}) = S_X(u.w_1 \dots w_i).w_{i+1} = u'.w_1 \dots w_i.w_{i+1}$. Eventually $S_X(u.w) = u'.w$. $\square$

Proposition 3 (Conjugate of local is local). *If* $(F, R_\bullet)$ *is an reversible causal graph dynamics and* $(L, S_\bullet)$ *is a local dynamics, then* $(L', T_\bullet)$ *is a local dynamics, with*

(*i*) $L' = F^{-1} \circ L \circ F$ *and*

(*ii*) $T_X(u) = R'_{F^{-1}(L(F(X)))}(S_{F(X)}(R_X(u)))$,

where the function $R'_\bullet$ *is such that* $(F^{-1}, R'_\bullet)$ *is a causal graph dynamics.*

Proof. Boundedness and uniform continuity by composition. Next, suppose: L is local, r_0 is such that for all X, Y if $X^{r_0} = Y^{r_0}$ then $F^{-1}(X)^0 = F^{-1}(Y)^0$ (given by uniform continuity of F^{-1}), r_{2b_F} is such that for all X, Y if $X^{r_{2b_F}} = Y^{r_{2b_F}}$ then $F(X)^{2b_F} = F(Y)^{2b_F}$ (given by uniform continuity of F), $b_{F^{-1}}$ is the bound given by the bounded inflation lemma applied on F^{-1}, b_L is the bound given by the boundedness of L and r_L the radius of locality of L. In the two following points, we chose a radius r' as follow:

$$r' = b^{F^{-1}}(r_L + 2 + max(r_0, 2b_F, r_{2b_F}))$$

Consider $|u'| > r'$.
[(*i*)] Let us show that there exists $u \in X$ such that $L'(X)^0_{u'} = X^0_u$. By definition of F^{-1}, there exists $w \in LF(X)$ such that $R^{F^{-1}}_{LF(X)}(w) = u'$. By bounded inflation of F^{-1}, we have $|w| > r_L$ and thus by locality of L, there exists $w' \in F(X)$ such that $S_{F(X)}(w') = w$. Finally by reversibility of F there exists $u \in X$ such that $R^F_X(u) = w$, and thus $u' = T_X(u)$. Notice that we have that $|S_{F(X)}R^F_X(u)| > r_0 + r_L + 2$. Using lemma 9 with $t = r_0$, we have: $LF(X)^{r_0}_{S_{F(x)}R^F_X(u)} = F(X)^{r_0}_{R^F_X(u)} = F(X_u)^{r_0}$. By definition of r_0, $F(X_u)^{r_0} = LF(X)^{r_0}_{S_{F(x)}R^F_X(u)}$ implies $X^0_u = F^{-1}(LF(X)^{r_0}_{S_{F(x)}R^F_X(u)})^0$, which leads by shift-invariance of F^{-1} to $X^0_u = F^{-1}(LF(X)^{r_0})^0_{R^{F^{-1}}_{LF(X)}S_{F(x)}R^F_X(u)}$. Hence $X^0_u = L'(X)^0_{u'}$.

[(ii)] Consider u as above and $v \in X_u^0$.

$$\begin{aligned}
T_X(u.v) &= R^{F'}_{LF(X)} S_{F(x)}(R^F_X(u.v)) \\
&= R^{F'}_{LF(X)} S_{F(x)}(R^F_X(u).R^F_{X_u}(v)) \text{ using shift-invariance of } F \\
&= R^{F'}_{LF(X)} S_{F(x)}(R^F_X(u)).R^F_{X_u}(v) \text{ because } |S_{F(X)} R^F_X(u)| > r_L + 2b^F + 2 \\
&= R^{F'}_{LF(X)}(S_{F(x)}(R^F_X(u))).R^{F'}_{LF(X)_{S_{F(X)}(R^F_X(u))}}(R^F_{X_u}(v)) \text{ using shift-invariance of } F^{-1} \\
&= T_X(u).R^{F'}_{LF(X)_{S_{F(X)}(R^F_X(u))}}(R^F_{X_u}(v))
\end{aligned}$$

We will now show that: $v = R^{F'}_{LF(X)_{S_{F(X)}(R^F_X(u))}}(R^F_{X_u}(v))$. Since $|R^F_{X_u}(v)| < 2b_F$ by bounded inflation of F, it is enough to show:

$$R^{F'}_{LF(X)_{S_{F(X)}(R^F_X(u))}}{}^{2b_F}(R^F_{X_u}(v)) = v.$$

By definition of r_{2b_F}, we have that: if $X^{r_{2b_F}} = Y^{r_{2b_F}}$ then $R^{F'}(X)^{2b_F} = R^{F'}(Y)^{2b_F}$. Let us show that $LF(X)^{r_{2b_F}}_{S^{2b_F}_{F(X)}(R^F_X(u))} = F(X_u)^{r_{2b_F}}$. By applying lemma 9 with $t = r_{2b_F}$, $LF(X)^{r_{2b_F}}_{S^{2b_F}_{F(X)}(R^F_X(u))} = F(X)^{r_{2b_F}}_{R^F_X(u)}$ which, by shift-invariance of F, is equal to $F(X_u)^{r_{2b_F}}$. As a consequence,

$$R^{F'}_{LF(X)_{S_{F(X)}(R^F_X(u))}}{}^{2b_F}(R^F_{X_u}(v)) = R^{F'}_{F(X_u)}{}^{2b_F}(R^F_{X_u}(v)) = R^{F'}_{F(X_u)}(R^F_{X_u}(v)) = v$$

by definition of $R^{F'}_{\bullet}$ $\square$

Second, we give ourselves a little more space so as to mark which parts of the graph have been updated, or not.

Definition 35 (Marked pointed graphs modulo). *Consider the set of pointed graphs modulo $\mathcal{X}_{\Sigma,\Delta,\pi}$ with labels in Σ, and ports in π. Let $\Sigma' = \Sigma \times \{0,1\}$ and $\pi' = \pi \times \{0,1\}$. We define the set of* marked pointed graphs modulo $\overline{\mathcal{X}}_{\Sigma',\Delta,\pi'}$ *to be the subset of $\mathcal{X}_{\Sigma',\Delta,\pi'}$ such that:*

- *$\forall u \in X$, if u is labelled with (x,a) and $\{u{:}(i,b), v{:}(j,c)\} \in X$, then $a = c$.*
- *$\forall v \in X$, if $\{u{:}(i,b), v{:}(j,c)\} \in X$ and $\{u'{:}(i',b'), v{:}(j,c')\} \in X$, then $u = u'$.*

Definition 36 (Mark operation). *We define the mark operation μ as the following local dynamics $(L^\mu, S^\mu_\bullet)$ over $\mathcal{X}_{\Sigma',\Delta,\pi'}$. For any X in $\mathcal{X}_{\Sigma',\Delta,\pi'}$:*

- *if the label of ε is (x,a) in X then its label is $(x,1-a)$ in $L^\mu(X)$.*
- *if $\{\varepsilon\!:\!(x,a),\varepsilon\!:\!(y,b)\} \in X$ then $\{\varepsilon\!:\!(x,1-a),\varepsilon\!:\!(y,1-b)\} \in L^\mu(X)$.*
- *if $\{\varepsilon : (x,a), v : (y,b)\} \in X$ with $v \neq \varepsilon$ then $\{\varepsilon : (x,a), v : (y,1-b)\} \in L^\mu(X)$,*
- $S^\mu_X(u) = \begin{cases} \varepsilon & \text{if } u = \varepsilon \\ (x,a)(y,1-b) & \text{if } \{\varepsilon\!:\!(x,a), v\!:\!(y,b)\} \in X \text{ with } v \neq \varepsilon \\ S^\mu_X(v).pq & \text{if } u = v.pq \text{ with } p,q \in \pi' \end{cases}$

and leaving the rest of the graph X unchanged.

Notice that the set of marked graphs $\overline{\mathcal{X}}_{\Sigma',\Delta,\pi'}$ is nothing but the subset of $\mathcal{X}_{\Sigma',\Delta,\pi'}$ obtained by as closure of μ, and shifts, upon $\mathcal{X}_{\Sigma\times\{0\},\Delta,\pi\times\{0\}}$. Notice also that $\overline{\mathcal{X}}_{\Sigma',\Delta,\pi'}$ is a compact subset of $\mathcal{X}_{\Sigma',\Delta,\pi'}$.

It turns out that any reversible causal graph dynamics admits an extension that allows for these marks.

Definition 37 (Reversible extension). *Let $(F, R_\bullet)$ be an reversible causal graph dynamics over $\mathcal{X}_{\Sigma,\Delta,\pi}$. We say that $(F', R'_\bullet)$ is a* reversible extension *of $(F, R_\bullet)$ if it is an reversible causal graph dynamics over $\overline{\mathcal{X}}_{\Sigma',\Delta,\pi'}$ such that,*

- *For any $X \in \mathcal{X}_{\Sigma,\Delta,\pi}$ and $u \in X$:*
  $$\begin{array}{lclclcl} F'(X\times\{0\}) & = & F(X)\times\{0\} & \quad & R'_{X\times\{0\}}(u\times\{0\}) & = & R_X(u)\times\{0\} \\ F'(X\times\{1\}) & = & X\times\{1\} & & R'_{X\times\{1\}}(u\times\{1\}) & = & u\times\{1\} \end{array}$$
- *For any $X \in \overline{\mathcal{X}}_{\Sigma',\Delta,\pi'}$ such that $|X| \leq p$ and $X \notin \mathcal{X}_{\Sigma\times\{0\},\Delta,\pi\times\{0\}}$:*
  $$\begin{array}{lclclcl} F'(X) & = & X & \quad & R'_X & = & u \mapsto u \end{array}$$

where p is that of theorem 3; $X\times\{0\}$ consists in pairing with 0 all the vertex-states and edges of X; and $u\times\{0\}$ is defined as $\varepsilon\times\{0\} = \epsilon$ and $u.ab\times\{0\} = (u\times\{0\})\!:\!(a,0)(b,0)$.

Proposition 4 (Reversible extension). *Any RCDG $(F, R_\bullet)$ over $\mathcal{X}_{\Sigma,\Delta,\pi}$ admits a reversible extension $(F', R'_\bullet)$ over $\overline{\mathcal{X}}_{\Sigma',\Delta,\pi'}$.*

To prove this result, we need to introduce two notions of projection of a marked graph. In definition 38 and lemma 10, $G(X)$, with $X \in \mathcal{X}_{\Sigma,\Delta,\pi}$ stands for the canonical representative of X in $\mathcal{G}_{\Sigma,\Delta,\pi}$, i.e. a classical graph having the same structure as X. This operations is detailed in appendix A.

Definition 38 (Upper and lower projections). *Let G be a graph in $G(\overline{\mathcal{X}}_{\Sigma',\Delta,\pi'})$. We define ${}^{\downarrow}G$ (resp. ${}^{\uparrow}G$) the lower (resp. upper) projection of G as the set of the connected component obtained after removing all marked vertices (resp. all non-marked vertices without used marked ports).*

Lemma 10 (Characterization of connected components). *Given G in $G(\overline{\mathcal{X}}_{\Sigma',\Delta,\pi'})$, the elements of the sets ${}^{\downarrow}G$ and ${}^{\uparrow}G$ are of the form $u.Y$ with $u \in \tilde{G}$ and $Y \in G(\overline{\mathcal{X}}_{\Sigma',\Delta,\pi'})$.*

Notice that the following proof uses the notion of local rules.

Proof. Let us construct such a reversible extension F'. Let p be that of theorem 3. For all $|X| \leq p$ and $X \notin \mathcal{X}_{\Sigma\times\{0\},\Delta,\pi\times\{0\}}$, we let $F'(X) = X$. We now assume that $|X| > p$. Given $L : \mathcal{X}_{\Sigma,\Delta,\pi} \to \mathcal{X}_{\Sigma,\Delta,\pi}$, we define $L^\star$ as the function $G \circ F$. Now for all $X \in \overline{\mathcal{X}}_{\Sigma',\Delta,\pi'}$, we define $F'(X)$ as the equivalence class modulo isomorphism of the following graph pointed on ε:

$$\left[\bigcup_{C \in {}^{\uparrow}G} C\right] \cup \left[\bigcup_{u.Y \in {}^{\downarrow}G} u.F^\star(\tilde{Y})\right]$$

where $G = G(X)$. Notice that if $G \in \mathcal{G}_{\Sigma\times\{0\},\pi\times\{0\}}$ then ${}^{\uparrow}G$ is empty and ${}^{\downarrow}G$ contains a single connected component $\varepsilon.G$ (the graph itself), thus F' computes F. On the other hand, if $G \in \mathcal{G}_{\Sigma\times\{1\},\pi\times\{1\}}$ then ${}^{\downarrow}G$ is empty and ${}^{\uparrow}G$ contains $\varepsilon.G$ only, thus F' computes the identity. Hence this F' is a good candidate for being a reversible extension of F. It remains now to check that F' is causal, vertex-preserving and reversible.
[*Causal*] Shift-invariance, boundedness and continuity follow directly from the shift-invariance, boundedness and continuity of both F and the identity.
[*Reversible*] Replace F by F^{-1} in the previous definition. □

In order to obtain our circuit-like form for reversible causal graph dynamics, we will proceed by reversible, local updates.

Definition 39 (Conjugate mark). *Given a reversible extension $(F', R'_\bullet)$ over $\overline{\mathcal{X}}_{\Sigma',\Delta,\pi'}$, we define the* conjugate mark K *as a dynamics $(L^K, S^K_\bullet)$ over*

$\overline{\mathcal{X}}_{\Sigma',\Delta,\pi'}$ as follows:

$$L^K = F'^{-1} \circ L^\mu \circ F' \quad \textit{and} \quad S^K_X(u) = T_{F'^{-1}(L^\mu(F'(X)))}(S^\mu_{F(X)}(R'_X(u)))$$

where the function $T_\bullet$ is such that $(F'^{-1}, T_\bullet)$ is a CDG.

Notice that by Proposition 3, the local update blocks are local operations. Moreover, since they are defined as a composition of invertible dynamics, so they are. In order to represent the whole of an reversible causal graph dynamics, it suffices to apply these local update blocks at every vertex.

Theorem 5 (Reversible localizability). *For any reversible causal graph dynamics $(F, R_\bullet)$ over $\mathcal{X}_{\Sigma,\Delta,\pi}$, $(F, R_\bullet)$ and $(\prod \mu)(\prod K)$ act the same on all but a finite number of graphs, where K is the conjugate mark of a reversible extension of F.*

Proof. By theorem 3, there exists $p > 0$ s.t. if $|X| > p$, R_X is invertible. On these graphs, the action of $(\prod \mu)(\prod K)$ is equivalent to $(\prod \mu)(F'^{-1}, R_\bullet^{-1})(\prod \mu)(F', R_\bullet)$. Therefore, given X s.t. $|X| > p$, we have $X \times \{0\} \mapsto (F', R_\bullet)F(X) \times \{0\}$ $\mapsto \prod \mu F(X) \times \{1\} \mapsto (F'^{-1}, R_\bullet^{-1})F(X) \times \{1\} \mapsto \prod \mu F(X) \times \{0\}$. □

Notice that for the finite number when the decomposition of theorem 5 does not apply, F is bijective. Therefore it just permutes those cases. Thus, this theorem generalizes the block decomposition of reversible cellular automata, which represents any reversible cellular automata as a circuit of finite depth of local permutations. Here, the mark μ and its conjugate K are the local permutations. The circuit is again of finite depth, a vertex u will be attained by all those K that act over $X^{r'}_u$, where r' is the locality radius of K. Therefore, the depth is less than $|\pi|^{r'}$. An example of such a decomposition is described in Figure 4.5. Moreover, it is interesting to notice that the construction proposed in this result is somehow simple than the corresponding construction for reversible cellular automata. Indeed, our marked graphs simply consist in the original graphs with an additional bit of information on each vertex and each port. The classical construction proposed in [**?**] changes the set of internal states Σ of each cell into the set Σ^2, which is much bigger. This is due to the fact that causal graph dynamics are defined on every possible graph of our configuration space, allowing us to simply remove the concerned vertex. For classical cellular automata, one must replace the cell by another one instead of just removing it.

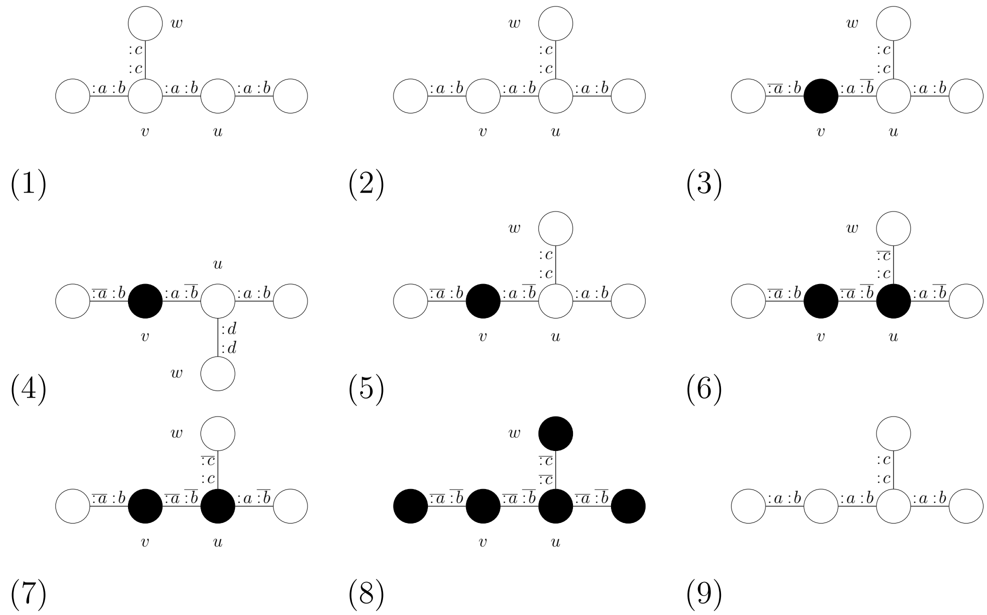

Figure 4.5: Block representation of the moving head dynamics. (1) Initially, no vertices are marked. (2) to (4) Application of K_v. First F is applied, then v is marked, followed by the application of F^{-1}. (5) to (7) Application of K_u. (8) The graph once every K have been applied. The vertices just need to be unmarked by the μ's. (9) Altogether this implements one time step of F.

4.3 Lifting the curse of vertex-preservingness

The curse of vertex-preservingness. When we first started studying reversible instances of causal graph dynamics, we assumed that we had a perfect example of such a dynamics. In [HM98], Meyer *and al.* introduced a model of lattice gas, i.e. particles moving and interacting on a lattice, where particles collisions induced local geometrical changes in the lattice. This example, at least in their ad hoc formalism, seems to be reversible. Figure 4.6 provides more details on this example. However, in section 4.1, we proved that any reversible causal graph dynamics preserves the size of all large enough graphs, which implies that this example can not be reversible in our model.

And what if all graphs are infinite? Vertex-preservingness, by definition, only concerns finite graphs. Let us imagine that every vertex of our graph is

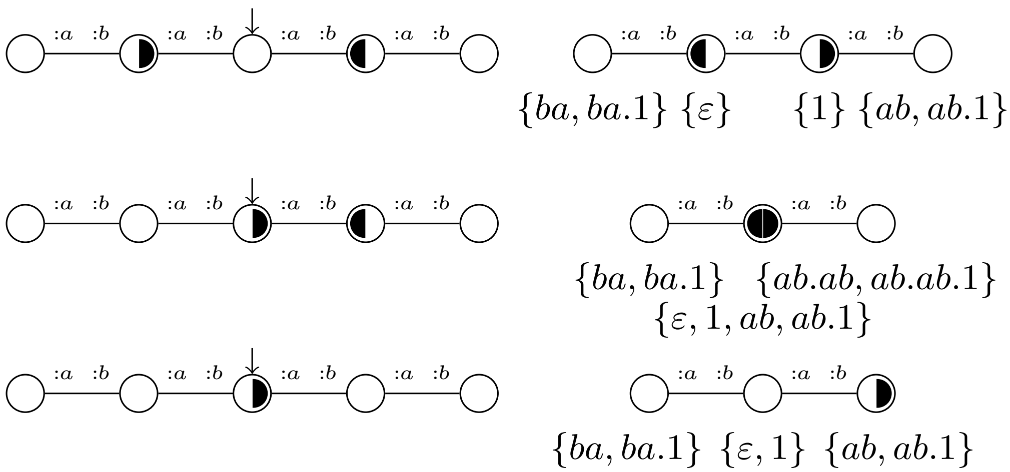

Figure 4.6: Lattice gas automaton with dynamical geometry. Vertices are arranged in lines of alternating port ab. Each vertex can carry travelling particles, moving left or right (the black half disks). Only three different neighbourhood are depicted here, as the others do not bring any additional information. (1) When two particles meet on a vertex, the vertex is split in two new vertices. (2) When two particles cross along an edge, the two vertices concerned by the crossing are merged. (3) When alone, the particle simply move left or right one step at a time. Notice that, in order to allow the possible duplication of a vertex, vertices in the right hand side graphs might have composed names of the form $\{u, u.1\}$, or simple names $\{u\}$ or $\{u.1\}$. This implies that the corresponding $R_\bullet$ operator is not bijective.

connected to an infinite "pool" of hidden vertices. Now, creating or deleting a new vertex simply comes down to extracting a vertex out of the pool, or hiding one inside it (see Figure 4.7). That being said, one can imagine going even further and altering the model itself in order to account for the presence of such an invisible matter without having to restrict ourselves to infinite graphs. This is what we call the "dark matter" approach.

Configurations. We define $\mathcal{Y}_{\Sigma,\Delta,\pi}$ to be $\mathcal{X}_{\Sigma,\Delta,\pi} \times (\pi^* \cup \omega)$. These are pointed graphs modulo, together with a translation of the pointer. The translation is specified by the sequence of ports along which the pointer ought to be moved. The element ω is short for all infinite chains of π^ω, as these will be

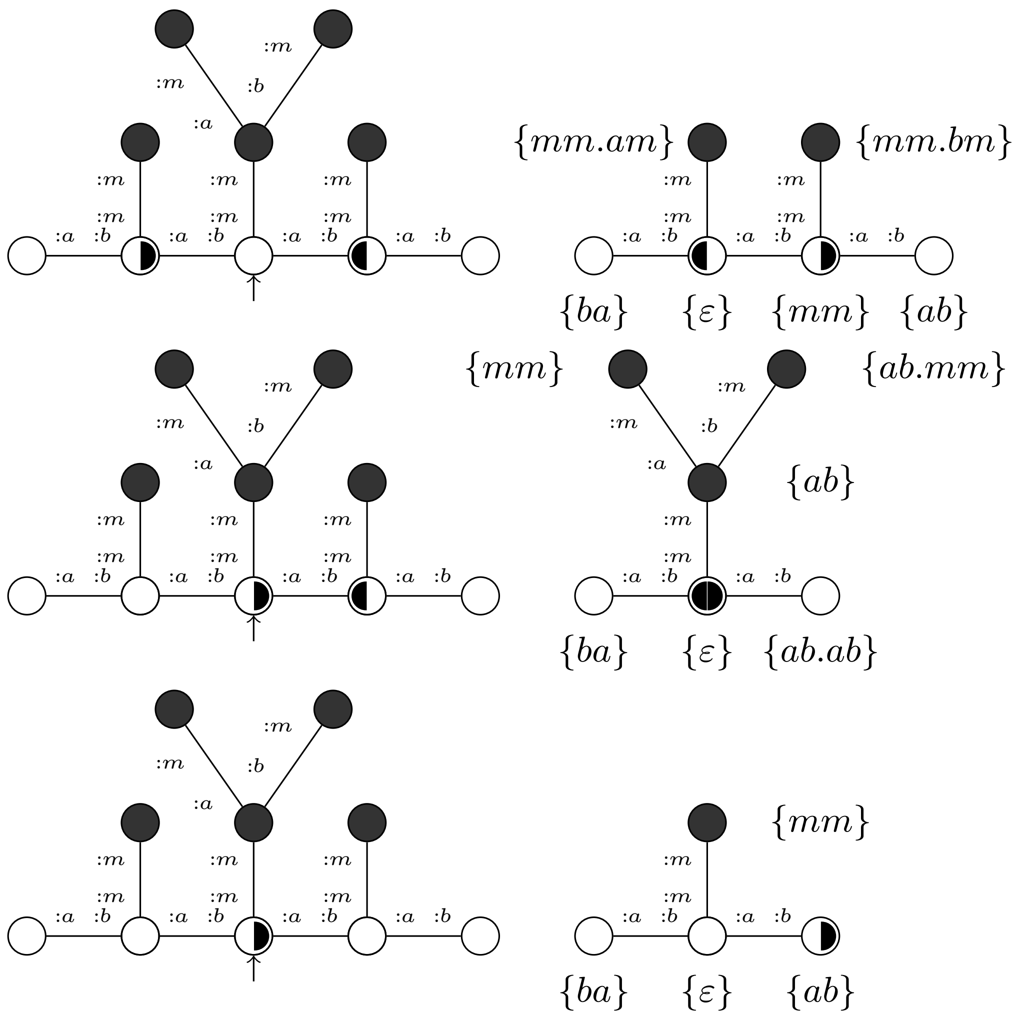

Figure 4.7: Lattice gas automaton with dynamical geometry, with dark matter. This time, each vertex possess an infinite tree of fresh vertices (in grey). The rule are almost the same, except the fact that the new vertex in (1) is extracted out of the dark matter, and the deleted vertex of (2) is now hidden in the dark matter. Notice that in this new rule, all vertices in the right hand side graphs have simple names (implying the bijectivity of the associated $R_\bullet$ operator).

treated in the same manner.

Equivalent configurations. Notice that the specified translation is arbitrary and thus not necessarily a path of the graph. If it is part of the graph, then the pointed-graph-modulo-with-translation naturally reduces to the corresponding translated pointed-graph-modulo-without-translation. More generally, if $u \in X$ then

$$(X, u.v) \to (X_u, v).$$

On the other hand, if $a \in \pi$ is not in X, then $(X, a.v)$ cannot be reduced and is said to be normalized. Clearly, every pointed-graph-modulo-with-translation Y admits a unique normalized form $Y{\downarrow}$, obtained by reducing as much as possible. We say that two pointed-graph-modulo-with-translation Y and Y' are equivalent, and write $Y \overset{*}{\leftrightarrow} Y'$, if and only if $Y{\downarrow} = Y'{\downarrow}$.
Let E stand for $(\varnothing, \omega)$. We add that

$$(X, \omega) \to E \qquad (\varnothing, v) \to E.$$

Disks and metric over configurations. For X in $\mathcal{X}_{\Sigma,\Delta,\pi}$, we define the disk X^r as usual. This is extended to $\mathcal{Y}_{\Sigma,\Delta,\pi}$ as follows. Given $Y = (X, u)$, consider $Y{\downarrow} = (X', u')$. If $|u'| \leq r$ then Y^r is defined to be (X'^r, u'). Else it is just E. From this definition of disks, there immediately follows a notion of metric on $\mathcal{Y}_{\Sigma,\Delta,\pi}$. Namely, that Y and Y' are at a distance 2^{-r} if and only r is the greatest radius such that $Y^r = Y'^r$. Notice that the metric is compact.
Dynamics and translation invariance. A dynamics is given by a function $F' : \mathcal{Y}_{\Sigma,\Delta,\pi} \to \mathcal{Y}_{\Sigma,\Delta,\pi}$ such that

$$Y \overset{*}{\leftrightarrow} Y' \quad \text{implies that} \quad F'(Y) \overset{*}{\leftrightarrow} F'(Y').$$

It decomposes it as

$$F'(X, u) = (F(X), R_X(u)).$$

For comparison with respect to the previous formalism, R_X has been relaxed to act over the whole of $(\pi^* \cup \omega)$ and not just from the vertices X to those of $F(X)$. The intuition is still that R_X indicates how a translation u with respect to X turns into a translation u' with respect to $F(X)$. This point becomes relevant to express translation invariance. A dynamics is translation-invariant if and only if for all $(X, u.v)$ with $u \in X$ we have that

$$R_X(u.v) = R_X(u).R_{X_u}(v).$$

Continuity and causality. A dynamics F' is continuous if and only if it is continuous in the usual meaning of the term, i.e. with respect to the above-given metric on $\mathcal{Y}_{\Sigma,\Delta,\pi}$. Given that the metric is compact, this is equivalent to uniform continuity. Now, consider (X, v) and (X, v') normalized. We can see that uniform continuity implies that for all n, there exists an m such that $R_X(v)_{1\dots n}$ is a function of $v_{1\dots m}$. In order to leave the 'dark matter' untouched, we need to make this function trivial beyond a certain point. Hence, the definition that a dynamics F' is causal if and only if it is shift-invariant, continuous, and such that there exists b such that for all $(X, u.v)$ normalized with $|u| = b$, we have

$$R_X(u.v) = R_X(u).v.$$

4.4 Summary of results

We managed to generalize two fundamental results of cellular automata theory to the model of causal graph dynamics. We proved that any bijective causal dynamics admits an inverse causal dynamics, and that any reversible causal dynamics admits a representation as a bounded depth circuit of local reversible operations. Moreover, we proved that reversible causal dynamics preserves the size of large enough graphs.

Chapter 5

Intrinsic universality

Intrinsic universality. When considering the problem of intrinsic simulation inside a model of computation such as Turing Machines or cellular automata, or in between models, the problem of qualifying the "structure" of the computation arises. Indeed, intrinsic simulation is about simulating an instance of a model while preserving the "structure" of the computation. In the case of cellular automata for instance, where this type of universality has been intensively studied [GMRT11, BT09, DL09, Oll08, Mar07], it is required that one must be able to obtain the simulated configuration by grouping cells of the simulating configuration, encoding a simulated cell into a block of fixed dimensions of simulating cells. In the case of causal graph dynamics, there is no such notion as "grouping" cells, due to the heterogeneity of the configuration. However, another way of interpreting a cell grouping in CA is to look at the global transformation used to encode the simulated configuration. It is of the form $E : \Sigma_1^{\mathbb{Z}} \to \Sigma_2^{\mathbb{Z}}$ where Σ_1 is the set of cells state of the simulated automaton and Σ_2 is the set of cells state of the simulating automaton, and, as the grouping is homogeneous and local, it is continuous (with respect to the metrics over $\Sigma_1^{\mathbb{Z}}$ and $\Sigma_2^{\mathbb{Z}}$) and shift-commuting (in an extended way). In fact it is just a CA from one set of configurations to another, possibly different, set of configurations. This is how we will proceed to define our notion of intrinsic simulation and universality.

This chapter. In this chapter, after defining a notion of intrinsic universality for causal graph dynamics, we present the construction of a family of intrinsically universal local rules. Due to the complexity of the construction of this family, it is impossible to provide a formal proof of its universality.

This construction is followed by the definition of an universal construction machine, describing a machine that is able to read as an input a description of a local rule together with a graph and to construct the initial state of a simulation of the application of the input local rule over the input graph.

The content of this chapter is based on [MM13] and [MM15], both co-authored with Bruno Martin.

5.1 Intrinsic simulation and universality

Even though, we only defined continuity and shit-invariance for transformations from the set of generalized Cayley graphs $\mathcal{X}_{\Sigma,\Delta,\pi}$ to itself, these definitions can be slightly altered to characterize continuity and shift-invariance for a transformation from a set $\mathcal{X}_{\Sigma_1,\Delta_1,\pi_1}$ to a set $\mathcal{X}_{\Sigma_2,\Delta_2,\pi_2}$. Indeed, defining the continuity of such a transformation is immediate using the two appropriate metrics over $\mathcal{X}_{\Sigma_1,\Delta_1,\pi_1}$ and $\mathcal{X}_{\Sigma_2,\Delta_2,\pi_2}$. Moreover, both definitions of shift-invariance and boundedness do not make any assumption on the image set of graphs on the considered transformation.

Definition 40 (Intrinsic simulation). *A localizable dynamics $(\mathcal{X}_{\Sigma_1,\Delta_1,\pi_1}, f_1)$ intrinsically simulates another localizable dynamics $(\mathcal{X}_{\Sigma_2,\Delta_2,\pi_2}, f_2)$ if and only if there exists a continuous, shift-invariant, bounded, injective, locally computable function $E : \mathcal{X}_{\Sigma_2,\Delta_2,\pi_2} \to \mathcal{X}_{\Sigma_1,\Delta_1,\pi_1}$ and a constant $\delta \in \mathbb{N}$ such that, for all graph $X \in \mathcal{X}_{\Sigma_2,\Delta_2,\pi_2}$:*

$$E \circ F_2(X) = F_1^{\delta} \circ E(X)$$

where δ corresponds to the number of steps needed to simulate one time step of F_2.

Now the definition of intrinsic universality comes naturally:

Definition 41 (Intrinsic universality). *A localizable dynamics $(\mathcal{X}_{\Sigma_u,\Delta_u,\pi_u}, f)$ is intrinsically universal if and only if, it intrinsically simulates any other localizable dynamics $(\mathcal{X}_{\Sigma,\Delta,\pi}, f)$.*

5.2 Preliminary results

Lemma 11 and lemma 12 are used to restrict the set of local rules we need to simulate in our construction.

Lemma 11 (Radius 1 is universal). *Let f be a local rule of radius $r = 2^\ell$ over $\mathcal{X}_{\Sigma,\Delta,\pi}$. There exists a local rule f' over $\mathcal{X}_{\Sigma^{\ell+1},\Delta\cup\{\star\},\pi^r}$ of radius 1 such that $(\mathcal{X}_{\Sigma\times\{1,\ldots,\ell\},\Delta\cup\{\star\},\pi^r}, f')$ simulates $(\mathcal{X}_{\Sigma,\Delta,\pi}, f)$.*

Proof. Outline. Over the first $i = 1 \ldots \ell$ steps, each vertex will grow some ancillary edges to, in the end, reach all neighbours in its neighbourhood of radius r. More precisely, states of vertices are kept identical, whereas an ancillary edge with state $\star$ is added between any two vertices at distance 2. Moreover, the vertices count until stage ℓ. At this point, the neighbours that were initially at distance r have become visible at distance one. The local rule f can be applied, all ancillary edges are dropped, and all counters are reset. □

Lemma 12 (Label free is universal). *Let f be a local rule of radius r over $\mathcal{X}_{\Sigma,\Delta,\pi}$. There exists a local rule f' over $\mathcal{X}_{\varnothing,\varnothing,\pi\cup\Sigma\cup\Delta^{|\pi|}}$ such that $(\mathcal{X}_{\varnothing,\varnothing,\pi'}, f')$ simulates $(\mathcal{X}_{\Sigma,\Delta,\pi}, f)$, where $\pi' = \pi \cup \Sigma \cup \Delta^{|\pi|}$.*

Proof. Outline. The presence of a label $i \in \Sigma$ on a vertex will be encoded by the presence of a dangling vertex on port i of this vertex. In the same fashion, if an edge labelled with $j \in \Delta$ connects two ports $u : a$ and $v : b$, then vertex u will have a dangling vertex on port j in its a^{th} port component and v a dangling vertex in its b^{th} port component. Notice that not all graphs are valid encodings, e.g. if a vertex has a dangling vertex on port $i \in \Sigma$ and on port $j \in \Sigma$ at the same time. Nevertheless this encoding verifies all the require properties as it is injective, continuous and shift-invariant.

□

Notice that these two constructions are not incompatible. Composing the two in the right order leads to the fact that any local rule can be intrinsically simulated by a local rule of radius one with no labels. In other words, the subset of localizable dynamics of radius one with no labels is intrinsically universal.

Corollary 1 (Radius 1 label free is universal). *Let f be a local rule of radius r over $\mathcal{X}_{\Sigma,\Delta,\pi}$. There exists a local rule f' over $\mathcal{X}_{\varnothing,\varnothing,\pi'}$ of radius 1 such* *that* $(\mathcal{X}_{\varnothing,\varnothing,\pi'}, f')$ *simulates* $(\mathcal{X}_{\Sigma,\Delta,\pi}, f)$.

5.3 Construction of a family of universal rules

We will now describe a family of intrinsically universal local rules $(f_d, \mathcal{X}_{\Sigma_u,\Delta_u,\pi_u})_d$ such that f_d simulates all rules over $\mathcal{X}_{\varnothing,\varnothing,\pi}$ with $|\pi| = d$. More precisely, all of these universal rules will act upon the same set of graphs $\mathcal{X}_{\Sigma_u,\Delta_u,\pi_u}$, and will only differ in their radius. In order to define these rules, we are faced with several problems:

- Our universal rules all act upon a given set of graphs of bounded degree $|\pi_u|$. We need to be able to encode any graph of bounded degree into our set of graphs $\mathcal{X}_{\Sigma_u,\Delta_u,\pi_u}$. Subsection 5.3.1 tackles this issue and introduces an encoding of any graph of bounded degree in a graph of degree 3.

- There is an unbounded number of local rules of radius 1 with no labels. Hence, the information of which local rule is to be simulated can not be stored as a label in Σ_u. Subsection 5.3.2 offers an encoding of any local rule in a subgraph whose purpose is to be attached to every simulated vertex.

- In order to simulate more than a single time step of the local rule, we must be able to create several instances of the graph containing its encoding and transmit these instances to the descendants of the simulated vertex. Subsection 5.3.3 offers a way to duplicate a subgraph describing a local rule, together with some synchronization tools.

A description of the functioning of the universal local rule is given in subsection 5.3.3. In this section we might refer to simulated vertices as "meta"-vertices since each of these vertices will be encoded in a graph structure.

5.3.1 Graph encoding

We choose the following encoding to represent any graph of bounded degree π in a graph of degree 3. For readability purpose, we will give explicit names to the 3 ports used in the following definition. The set of ports in the encoding can be assimilated to $\{0, 1, 2\}$. The three port are: *previous*, *next* and *neighbour*. The set containing those three ports will be referred to as π_{graph}. We define the set of labels Σ_{graph} as the set $\{VERTEX,PORT\}$.

Definition 42 (Graph encoding). *Given a set of ports π, consider the transformation $E_\pi^{graph} : \mathcal{X}_\pi \to \mathcal{X}_{\Sigma_{graph},\pi_{graph}}$ defined as follows:*

- *To each vertex v in X, corresponds $\pi+1$ vertices $v_0, ..., v_\pi$ in $E_\pi^{graph}(X)$ and the following edges: for all $i \in \{0, ..., |\pi|\}$, $\{v_i : next, v_{i+1} : previous\}$. v_i has label PORT for $i < |\pi|$ and $v_{|\pi|}$ has label VERTEX.*
- *To each edge $\{u : i, v : j\}$ in X corresponds an edge $\{u_i : neighbour, v_j : neighbour\}$.*

The idea is to split the encoded vertex into $|\pi| + 1$ vertices and arrange them into a ring. Each vertex v_i for $i < |\pi|$ represents a port of the encoded vertex. The last vertex v_π is here to mark the beginning of the ring (the vertex representing port 0 will be found on its port *next*). Figure 5.1 describes the encoding for graphs with $|\pi| = 3$.

Lemma 13 (E_π^{graph} is a good encoding). *Given π, E_π^{graph} is continuous, shift-invariant and injective.*

Proof. The proof of this result is pretty straightforward. As E_π^{graph} acts locally on the graph, continuity and shift-invariance are instantaneous. Moreover, any change in the original graph will result in a change in the encoded graph as all information on the topology is preserved.

□

5.3.2 Local rule encoding

[**General structure**] We need to encode any local rule of radius 1 without label into a subgraph. A rule of degree $|\pi|$ can be seen as an array of fixed length (the number of possible neighbourhoods) containing all the possible outputs of the local rule. We choose to arrange all these outputs along a line graph together with a description of the corresponding neighbourhood. The description of the neighbourhoods is detailed in subsection 5.3.3. Figure 5.2 represents such an encoding for the local rule inducing the turtle dynamics.

[**Addresses and identification**] We also need to identify a meta-vertex of an output to another meta-vertex in another output, in order to proceed to a

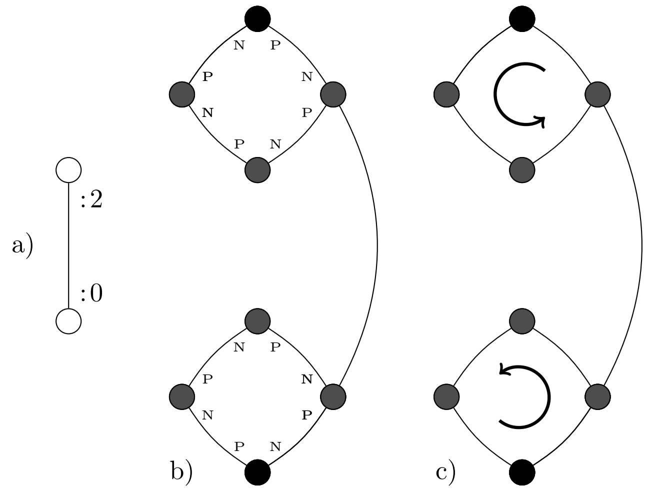

Figure 5.1: In this example, $\pi = \{0, 1, 2\}$. a) represents a graph composed of two vertices connected through ports 2 and 0. b) represents the encoding of this graph. Each vertex is represented by 4 vertices forming a ring. The darkest vertices have label VERTEX while the grey vertices have label PORT. The ports used in the ring are next and previous (N and P). The edge linking the two vertices is represented by an edge between the third vertex of the first ring (representing port 2 of the first vertex) and the first vertex of the second ring (representing port 0 of the second vertex). Finally, c) presents a lighter representation of the same encoding where an arrow indicates the orientation of the rings. For the sake of clarity, this latter representation will be used in the following figures.

graph union. This is done by adding to each vertex labelled by *VERTEX* a line graph containing a path towards the other vertex. Figure 5.3 represents the graph encoding the turtle local rule with these addresses. Figure 5.4 represents the graph encoding the inflating line local rule.

[**Inheritors and disowned vertices**] Inside an output subgraph, there are two types of meta-vertices: The ones that need to receive a copy of the local rule graph and the others. We use a product label to mark the meta-vertices that will inherit of a copy of the local rule. In the example of the turtle, all

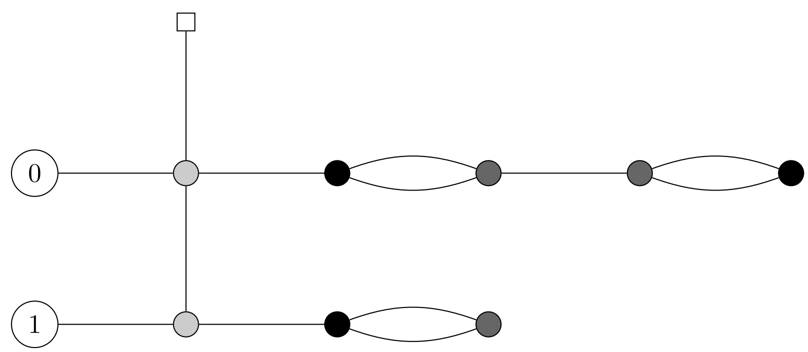

Figure 5.2: Encoding of the turtle rule. Black vertices are vertices labelled by VERTEX, dark grey vertices are labelled by PORT. Light grey vertices are part of line structure onto which all outputs are attached. Vertices on the left of the vertical line are labelled by bits and correspond to the number of the outputs in an enumeration of all possible neighbourhoods. We chose not to use the neighbourhood encoding used in 5.3.3, as there are only 2 different neighbourhoods. Finally the square vertex represents the top of the line structure.

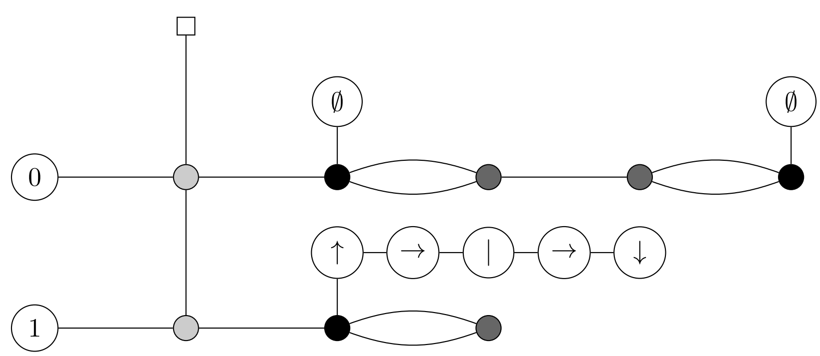

Figure 5.3: Encoding of the turtle rule including the addresses. The empty set label is used to specify that the meta-vertex does not need to be identified to another meta-vertex. When the address is not empty, it is encoded in a line graph using 4 different labels: $\uparrow$, $\rightarrow$, $\downarrow$ and $|$. $\uparrow$ indicates to move on the father meta-vertex. $\downarrow$ indicates to go down from a father meta-vertex to its output. $\rightarrow$ indicates to travel along the port NEXT in a meta-vertex. $|$ indicates to travel along the port neighbour between two meta-vertices.

meta-vertices are marked while in the example of the inflating line, only the meta-vertices having an empty address are marked.

5.3.3 Description of an intrinsically universal rule

Applying a local rule to every vertex in a graph consists in several stages:

(*i*) Each vertex observes its neighbourhood,

(*ii*) Each vertex deduces the output subgraph it has to produce according to the local rule,

(*iii*) A graph union of all these subgraphs is computed to produce the final graph.

The universal local rule implements those three stages, with an additional stage:

(*ii*)$*$ The encoding of the local rule is duplicated into each meta vertex of the chosen output subgraph.

Moreover, a universal local rule must synchronise the simulation in every meta-vertex in order to perform the graph union only when all subgraphs are chosen and all duplications are over.
We will detail how each of these stages are performed by the universal local rule.

[**Neighbourhood observation**] First the meta-vertex proceeds to generate a matrix of vertices of size $|\pi|+1$ to store the connectivity of its neighbourhood. A new vertex is attached to the vertex labelled *VERTEX* and starts moving along the ring of vertices labelled *PORT* growing the matrix in two passes. Figure 5.5 describes this growing process on a meta-vertex of degree 9.
Once the matrix is built, the machine vertex starts a depth first search (DFS) of depth 1 on the meta vertex it is attached on. It grows 2 edges (or arms) that will travel in the graph and four unary counters to keep track of the DFS status. The unary counters are line graphs of length $|\pi|+1$ and $|\pi|$. The two counters of length $|\pi|+1$ keep track of which meta-vertex can be found at the end of each arm while the two counters of length π keep track of the ports currently considered. Figure 5.7 represents the structure of a

counter, and figure 5.8 describes the structure used to store the current state of the DFS. While visiting a vertex u_i, its port p_j and a vertex u_k and its port p_l, an edge is created between cells (i, j) and (k, l) of the matrix if the edge $\{u_i : p_j, u_k : p_l\}$ is present in the graph. Once the DFS is over, the matrix contains enough information to determine the neighbourhood of the vertex. Figure 5.6 presents the two different matrices for neighbourhoods in graphs of degree 1.

Note on the description of the neighbourhood. The usage of a matrix to encode the neighbourhood of a meta-vertex is the most general solution we can implement. However, in most of the cases, we do not need that much information. In the two examples we develop in this section (the turtle and the inflating line), it is only required to test for the existence of a potential neighbour on each port of the meta-vertex, and the local rule does not require to know its complete connectivity. Hence, in both local rule encodings, we will use an ad hoc encoding of the neighbourhoods. For the turtle rule, we will use a single bit to represent the two possible neighbourhoods. For the inflating line, a string of bits is used. For each port of the vertex we proceed as follow: If there is no neighbour on this port we add a 0 to the string. If there is a neighbour on the port, we add a 1 to the string, followed by a 0 or a 1 depending on the port at the other end of the edge.
[**Choosing the output subgraph**] After recording the local connectivity, the machine vertex starts to travel down the local rule encoding and compare this recording to the information attached to each output, stopping when the two are matching.

[**Duplicating the local rule encoding**] The machine then initiates a DFS on the chosen output graph. The purpose of this DFS is to search for marked meta-vertices. During the DFS, every time a marked meta-vertex is met, the DFS is paused and a new DFS starts from the root of the local rule encoding. This new DFS will explore the local rule encoding while constructing a new copy of it. This can be done by maintaining a stack structure containing the path followed in the graph during the DFS. When the machine encounters an edge leading toward a previously visited vertex, it uses this stack to backtrack and find the right edge to create in the new version of the graph. These two DFS act on graphs of degree 3 and 4 and thus do not require the same counter structures as the DFS in the neighbourhood observation stage, as everything can be stored using a bounded number of labels in the vertices.

This new version of the local rule encoding is then attached to the marked meta-vertex and the first DFS is resumed.

[**Graph union and vertex identification**] Once the DFS is over, meta-vertices of the output graph start moving in the graph according to the addresses attached to them. Meanwhile, the local rule encoding is reduced to get rid of all the unused outputs, leaving the only chosen output attached to the simulated meta-vertex.

[**Merging two meta-vertices**] After moving according to its address, a meta-vertex will meet its target meta-vertex and they will try to merge. Notice that the target meta-vertex might also try to merge with a third vertex, and so on, forming a sequence of meta-vertices that must be merged in a single meta-vertex. This can lead to two very distinct situation:

- The sequence is not cyclic,
- The sequence forms a cycle.

In the first case, the first vertex of the sequence will perform the merging, followed by the second and so on until all the meta-vertices are merged as a single meta-vertex. In the second case, no meta-vertex can decide to start the merging process as every meta-vertex sees itself in the middle of the sequence. Meta-vertices can easily decide whether this is the case by growing a new edge whose extremity will travel along the sequence. If the edge reaches the end of the sequence, then the merging process will start. If not, a synchronization process will start.

[**Synchronization process**] If synchronization is required during the merging process, then we can assume that these meta-vertices are synchronized (i.e. they decide to start the merging process exactly at the same time). Indeed, if they were not synchronized, then the symmetry could have been broken in the local rule encoding, as only the neighbourhood of simulated meta-vertex has an influence on the time step at which meta-vertices of its local rule encoding decide to merge. Two problems now arise:

- In order for the cycle to collapse in a single meta-vertex in a single time step, the universal local rule must be able to "see" the whole cycle, hence its radius must be of at least half the length of the larger possible identification cycle.

- Meta-vertices are not composed of a single vertex. They contain at least $|\pi| + 1$ vertices and might also contain a local rule encoding. All these vertices need to be simultaneously merged with their corresponding vertices in the previous and next meta-vertices in the merging cycle.

The first problem is easy to solve as we are constructing a family of intrinsically universal local rules. A given local rule can only produce merging cycles of bounded length, hence will be simulated by one of our universal local rule.

The second problem can be solved using a solution of a problem known as the Firing Squad Synchronisation Problem (FSSP) over graph automata [RFH+72, Maz88]. The construction uses labels on the vertices of a graph in order to synchronize all the vertices using only local communication between vertices. Moreover, the solution only depends on the degree of the graph to synchronize. In our case, we need to synchronize meta-vertices and their local rule encodings, which are of bounded degree 4. The identification process will be performed as follows:

- Meta-vertices will detect that they are in a cycle of identification,
- Meta-vertices start a FSSP on their main vertex,
- The FSSP synchronizes every vertex composing the meta-vertex and its potential local rule encoding,
- While propagating the FSSP, new edges are built between vertices of the meta-vertex and their corresponding vertices in the previous and next meta-vertices.
- When all vertices are synchronized, the universal local rule performs a merging of the vertices, leading to a single meta-vertex and its local rule encoding.

Figures 5.9 and 5.10 describe the different possible case of merging sequences and the synchronization process. When all mergings are performed, the original meta-vertices are destroyed, leaving only the new graph and can be restarted to simulate the next time-step.

[**Overall synchronization and counter structures**] One last step of synchronization is required to allow faster meta-vertices to wait for slower meta-vertices. This is done by adding to each local rule encoding a counter structure. This counter structure implements a binary counter decreasing its value at each time step. When the value 0 is reached, all meta-vertices finalize the merging process and generate a new automaton to pursue the simulation. Notice that the counter structure is necessarily a (at least) binary structure, as it must be duplicated together with the local rule encoding. If the counter was of the unary type, it could require a linear time to copy it, hence it will necessarily finish counting before the end of the duplication.

Figures 5.11 and 5.12 describe the complete simulation of one time step of the turtle dynamics over the graph containing two vertices.

5.4 Building instances: Universal constructing machine

While this formalism of intrinsic universality captures the idea we have in mind of what would be nowadays called an interpreter, another less formal definition suggested by von Neumann, namely the universal construction machine, is closer to the notion of a compiler. Von Neumann's idea, directly inspired by Turing's universal machine, is that there must exist a machine (not necessarily a Turing machine) which, when provided a suitable description of an instance of a computational model, constructs a copy of it. This definition is particularly useful when considering the problem of self-reproduction [Arb88]. The most classical example to illustrate this definition is the uniform generation of Boolean networks where a Turing machine receives as an input the standard encoding of the circuit and its size and explicitely generates the corresponding Boolean network [BDG88]. Though detached from any mathematical formalism, this notion particularly fits to our model, in as much as we are allowed to modify the topology and have enough freedom to design such a machine inside the model itself. In our case we can imagine having a "machine" (in fact, just a vertex) reading two "tapes" (linear graphs) containing a description of a graph and a description of a local rule, i.e. of another graph encoding the local rule as described above. The goal is then to build the described graph and to attach to each of its vertex a local rule encoding.

5.4.1 Encoding the initial graph

The usual way to encode a graph into a linear structure is to perform a DFS on the graph while remembering the edges leading to any previously visited vertex. The alphabet we use to encode the initial graph of a dynamics of degree $|\pi|$ and of labels Σ is the following:

$$\mathcal{A}_\pi = \pi^2 \cup \{\$, ;, |\} \cup \Sigma$$

With Σ the finite set of labels of the vertices and $\{\$, ;, |\}$ some arbitrary symbols used as delimiters (we need 3 of them). Figure 5.13 gives an example of the encoding of a graph with $\pi = \{1, 2, 3\}$.

The string encoding the graph is a sequence of words, one for each vertex, describing the backward edges (*i*e the edges leading to one of the previously visited vertices) and the forward path leading to the next vertex visited by the DFS. The structure of a word is described as follow:

$$\$\sigma\ (i_1, j_1)\overbrace{|\ldots}^{n_1}(i_2, j_2)\overbrace{|\ldots}^{n_2}...; (s_1, t_1)(s_2, t_2)\ldots(s_n, t_n)\ \$\text{next word...}$$

where:

- \$ plays the role of word separator.
- $\sigma \in \Sigma$ is the label of the vertex.
- $(i, j)\overbrace{|\ldots}^{k}$ describes the existence of an edge from port i of the current vertex to port j of the k^{th} vertex when backtracking in the DFS.
- ; is a separator between the backward edges and the forward path.
- $(s_1, t_1)(s_2, t_2)\ldots(s_n, t_n)$, with $(s_i, t_i) \in \pi^2$, describes the path from the current vertex to the next vertex in the DFS.

The backwards edge $(i, j)\overbrace{|\ldots}^{k}$ is described using a unary description of the number of times a backtracking has to be made in the DFS. This unary encoding is here to simplify the functioning of the universal machine described in the next section and does not change the time complexity of the construction of the graph. Notice that, as all our graphs are generalized Cayley graphs, they are pointed and thus this encoding is unique (the root of the

DFS being the empty path ε). This encoding is not the only way to encode a graph in a string and maybe not the most efficient way but it conveniently fits to our needs while being easy to describe. We could have equivalently defined our encoding using a Breadth First Search algorithm instead of a DFS without changing the complexity of the encoding.

This encoding is both injective and computable (it simply consists in a DFS).

In the following, the encoding of a generalized Cayley graph X in a linear graph is written $\langle X \rangle$. Using the construction of the previous section, we can also define the encoding of a local rule f in a linear graph: $\langle f \rangle$. It consists in two successive encoding: first the local rule is encoded into a graph of $\mathcal{X}_{\Sigma_u,\Delta_u,\pi_u}$ which, in turn, is encoded into a linear graph.

5.4.2 Universal machine

The universal machine we design is implemented in the model itself. It consists in a single vertex to which the two encodings $\langle X \rangle$ and $\langle f \rangle$ are connected.

The universal machine can now read the string $\langle X \rangle$ describing X in order to build the graph. While building it, each vertex receives a copy of the local rule encoding. The universal machine is very similar to the machine performing the stage of duplication of the local rule encoding of the universal dynamics described in subsection 5.3.3. However, its functioning is simpler as the input consists in an adequate description of the graph to build instead of the graph itself.

The universal machine itself is a vertex of degree 7:

- Three edges for the inputs: one for $\langle X \rangle$ (read only) and two for $\langle f \rangle$ (read and top),
- Two edges for manipulating the graph being constructed (one pointing at the last added vertex and another travelling along the DFS tree to create backward edges),
- Two edges for manipulating a stack used to store the sequence of paths linking to consecutive vertices in the DFS (one pointing at the start of the stack, the other reading it),

The execution of the universal machine is described by a simple local rule acting as the identity everywhere except in the neighbourhood of the machine itself. The machine proceeds as follow:

- If the symbol \$ is read, followed by σ, it adds the label σ in the last added vertex,

- If $(i,j)\overbrace{|\ldots}^{k}$ is read, the machine uses the stack to backtrack k times in the DFS tree and add the edge (i,j) between the last added vertex and the vertex reached at the end of the backtracking.

- If the letter ; is read, all backward edges have been added,

- After letter ;, the machine read the (s_i, t_i) one by one, following the described path, whose last edge has to be created together with a new vertex.

5.5 Summary of results and open problems

In this work, we provide a definition of intrinsical simulation and intrinsical universality for causal graph dynamics. We then construct a family of intrinsically universal instances of this model. All the local rules of this family act on the same set of graphs and only differ in their radius.
This construction is in no manner optimal, and can still be optimized in various ways:

- One could achieve a similar result with a construction on graphs of smaller degree, and with a smaller label set,

- The time-complexity of the simulation can probably be decreased by optimizing the structure of the graph encoding and the local rule encoding. For instance one could imagine using a set structure to encode the local rule, changing a linear access time (in the number of possible neighbourhoods) in a logarithmic access time.

Moreover, it seems that this is the best result we can achieve with this kind of construction, as it seems impossible to construct a unique intrinsically universal rule.
(Non) existence of a single universal rule. The construction presented in section 5.3 describes a family of intrinsically universal local rules, and not an universal local rule. All local rules in this family act on the same set of graphs, and only differ in their radius. Having universal local rules with

arbitrary large radius is only required in the last part of the construction, for the merging process. When meta-vertices decide to merge into a single meta-vertex, and the merging sequence forms a cycle, the local rule must be able to either:

- order the meta-vertices and proceed to merge them one-by-one according to that order
- or "see" all the meta-vertices, synchronize them, and proceed to the merging in one time step.

The latter case is only possible if the radius of the local rule is large enough, and that is the solution we adopted here. In the former case however, we must order meta-vertices that are descendant of different vertices of the simulated graph. This requires to be able to unambiguously order the meta-vertices in any disk of radius one of our simulated graph. This is equivalent to have a clean coloration of the simulated graph. The coloration will then give us a way to totally order the descendants of the meta-vertices, and hence gives us a way to proceed to the merging without requiring any synchronization process.
However, to use a clean coloration, we must prove that any local rule can be modified to take the coloration into account and maintain it over time. This, in turn, requires to be able to locally break the symmetries in the image graph, which might be impossible for some graphs.
Hence, it seems impossible to construct a unique intrinsically universal rule, at least using this type of constructions.

Notice that, although we need a family of instances to simulate all the possible instances, most of the "natural" instances can be simulated by the universal rule that allows merging sequence of arbitrary length and only forbids merging cycles of length greater that 4, i.e. the universal local rule of radius 2.

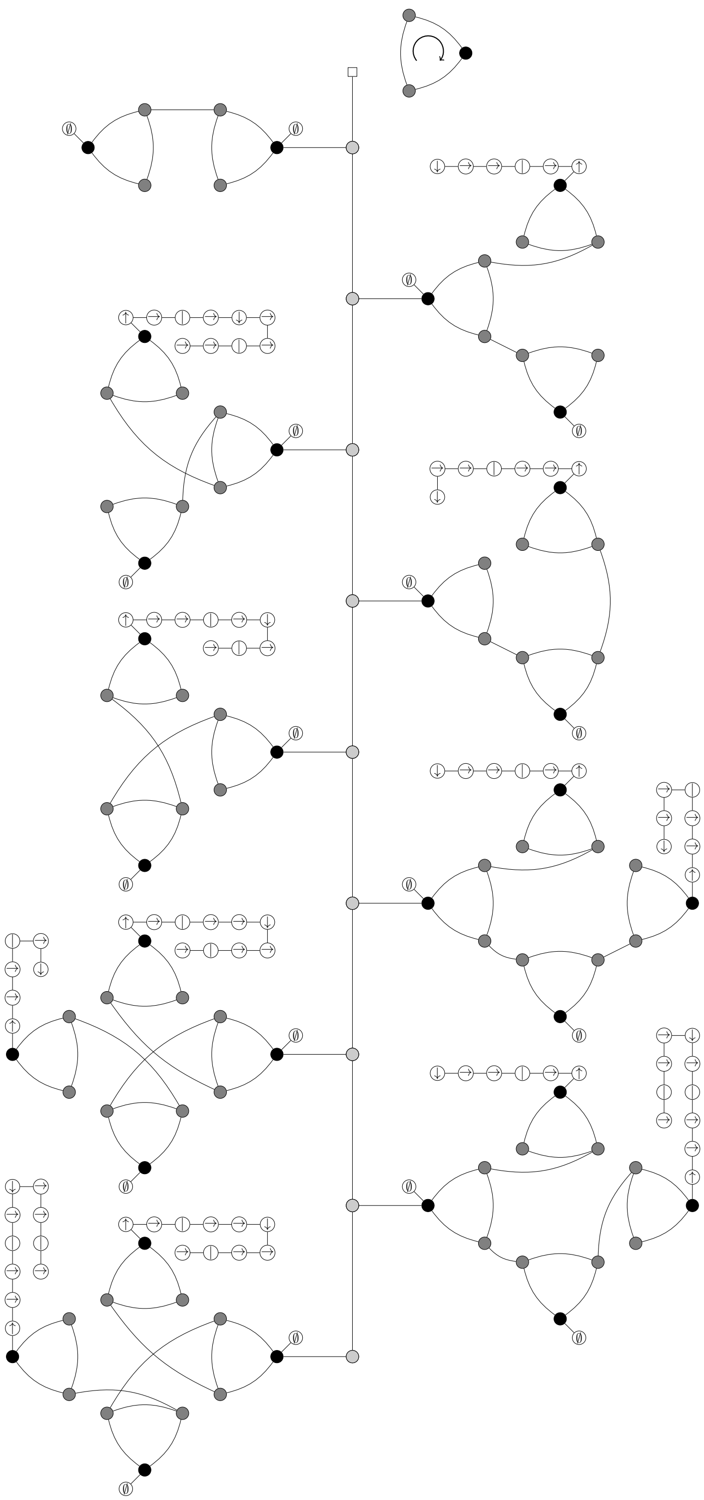

Figure 5.4: Encoding of the inflating line rule including the addresses. There exists 9 different neighbourhoods of radius 0 on graphs of degree 2, thus the presence of 9 different outputs in the encoding. Numbering of the possible outputs are neglected here as they do not bring any information to the understanding of this encoding.

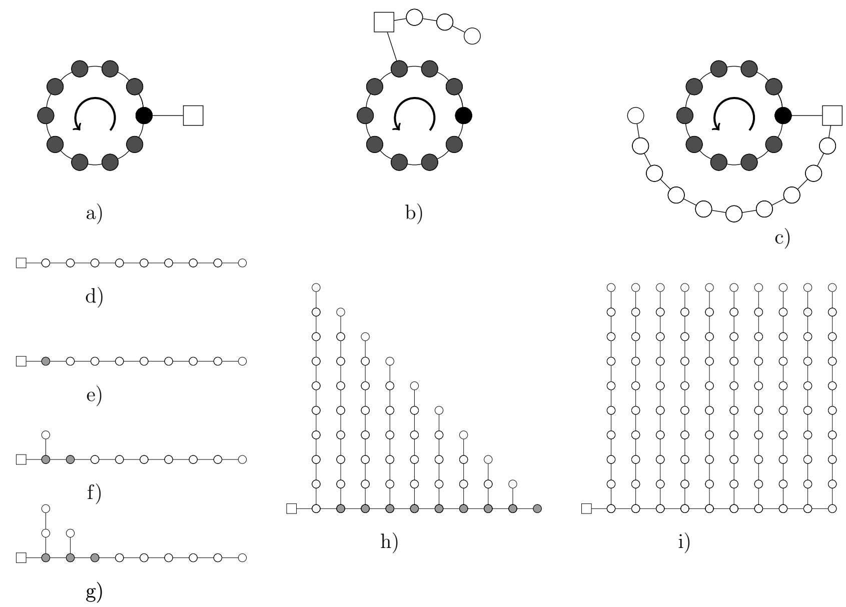

Figure 5.5: Growth of the connectivity matrix. A "machine" vertex starts to run along the ring and for each vertex it passes, adds a new vertex to a line graph: a) At first the line graph is empty and the machine vertex is attached on the VERTEX vertex. b) After three steps. c) After 10 steps, the machine vertex is back on the first vertex and start the second pass. d), e), f), g) represent the first 4 steps of the second pass. The machine sends a signal (in grey) that triggers the growth of each column while moving along the ring. h) represents the 11^{th} step where the signal reaches the last column and the machine arrives at the VERTEX label again. The machine sends an "end" signal to stop the growth of the columns. i) represents the final matrix (after 20 steps).

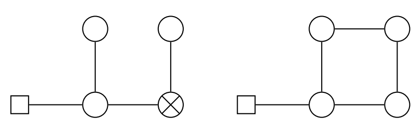

Figure 5.6: The two matrices encoding respectively the neighbourhood where no neighbour is present, and the neighbourhood where another vertex is present on the single port. In the first graph, the bottom right vertex is crossed to indicate that there is no “second” vertex in the neighbourhhod.

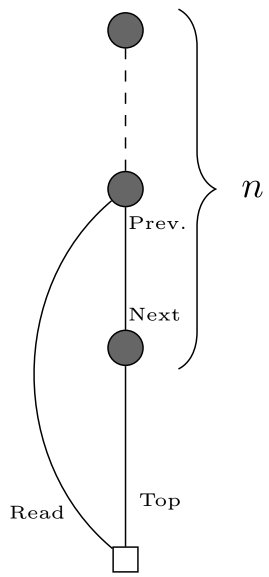

Figure 5.7: A counter structure. It consists in a line graph of the appropriate length. The origin of the line can grow an arm to read the counter, one vertex at a time. It is easy for an automaton to grow a counter of the appropriate size by running along a meta-vertex and generating a new vertex for each visited port (see matrix generation). All vertices composing the counter have the same label.

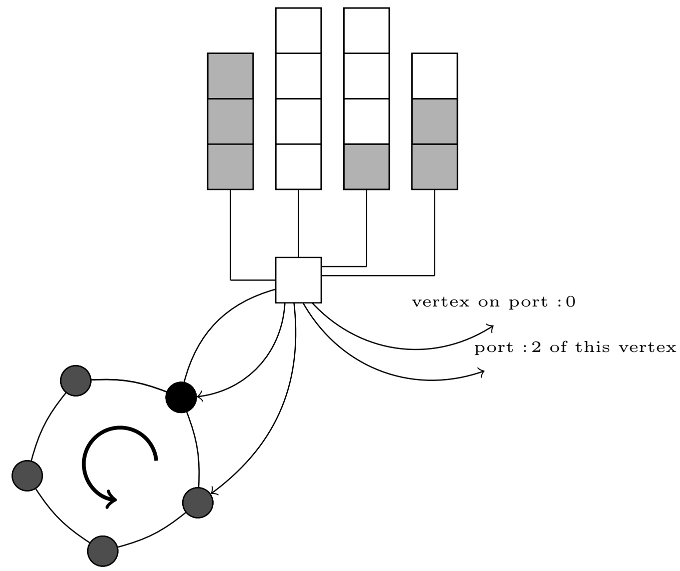

Figure 5.8: DFS structure for the neighbourhood exploration of a vertex of degree $|\pi| = 4$. At the top: 4 unary counters structure. The DFS explores the neighbourhood of the center vertex by considering every possible pair of vertices (including the center vertex, as there might be loops the graph). The two center counters are used to keep track of which vertices are currently being visited. The two smaller counters are here to keep track of which port is currently considered in each of these vertices. Here, the currently considered pair is composed of the "center" vertex and its neighbour of port : 0 and their ports :3 and :2.

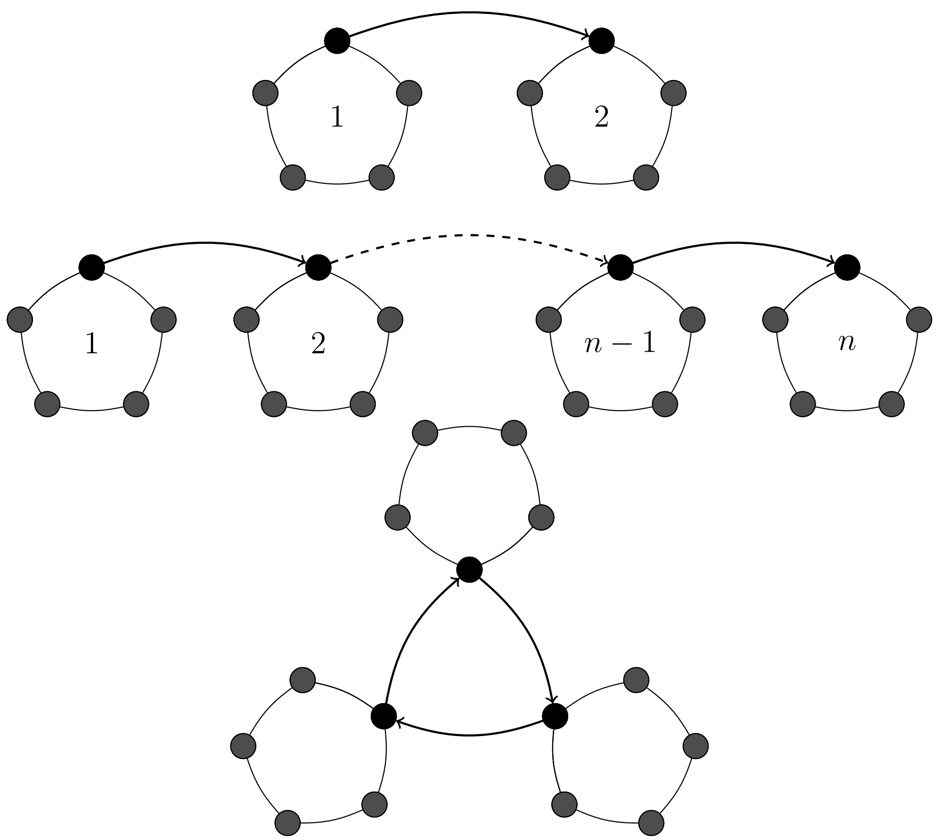

Figure 5.9: Different types of merging sequences. The two top cases are solved by ordering the vertices according to the sequence, and then having the first one merged to the second one, and so on. In the last case, the sequence forms a cycle, and a synchronization is necessary to perform the simultaneous merging of the cycle.

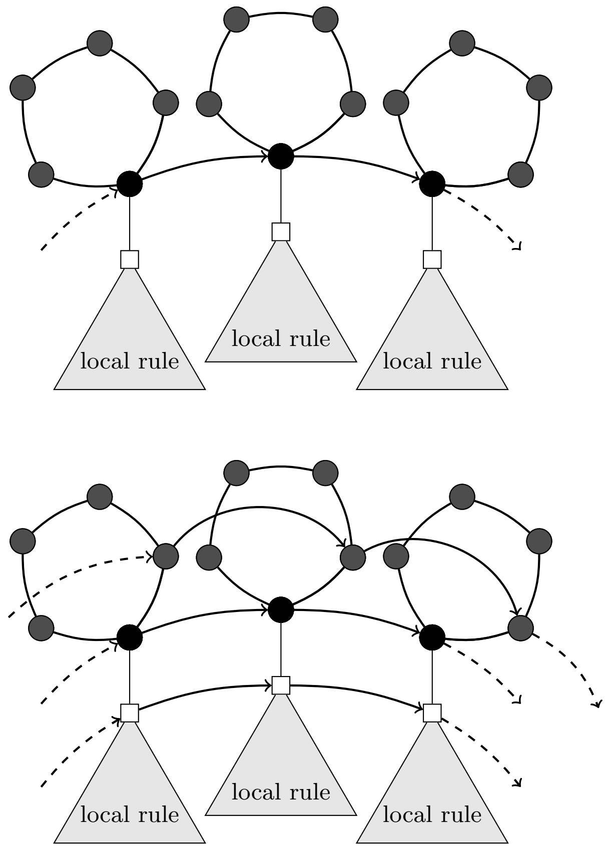

Figure 5.10: Synchronization process of three meta-vertices and their local rule encodings. All meta-vertices start a FSSP on their vertices. At the beginning, only the “main” vertex is connected along the merging cycle to the others “main” vertices (top graph). As the FSSP is propagated, the vertices connect themselves to their corresponding vertices in the previous and next vertices. The bottom graph describes the same graph, two propagation steps later. When the FSSP is completed, all vertices “fire” exactly at the same time and perform a merging along all the built cycles, resulting in a single meta-vertex and its local rule encoding.

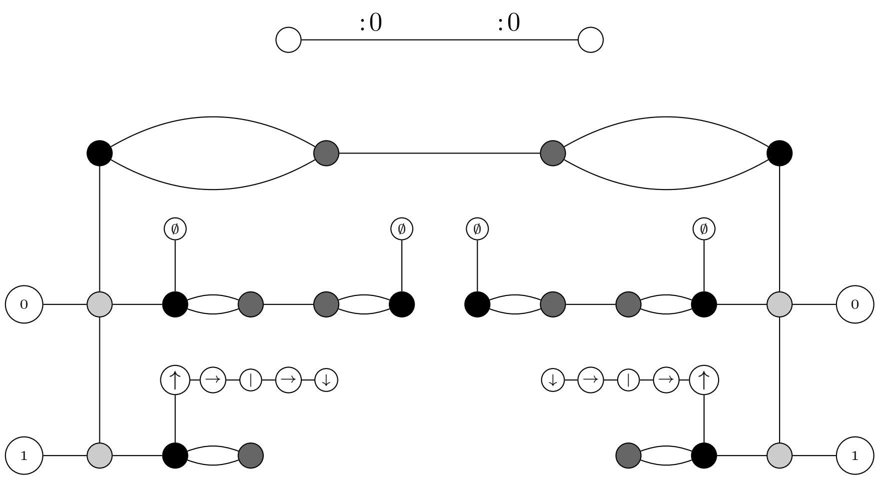

Figure 5.11: A graph of degree $|\pi| = 1$ and its encoding together with turtle local rule encodings. Each of the two meta-vertices receive a version the local rule encoding. Once again, the description of the different neighbourhoods in the local rule encoding is not made using matrices, as there are only two possible cases, instead we used single digits. Here the neighbourhood with only one single vertex is encoded by a 0 while the neighbourhood with two vertices is encoded by a 1.

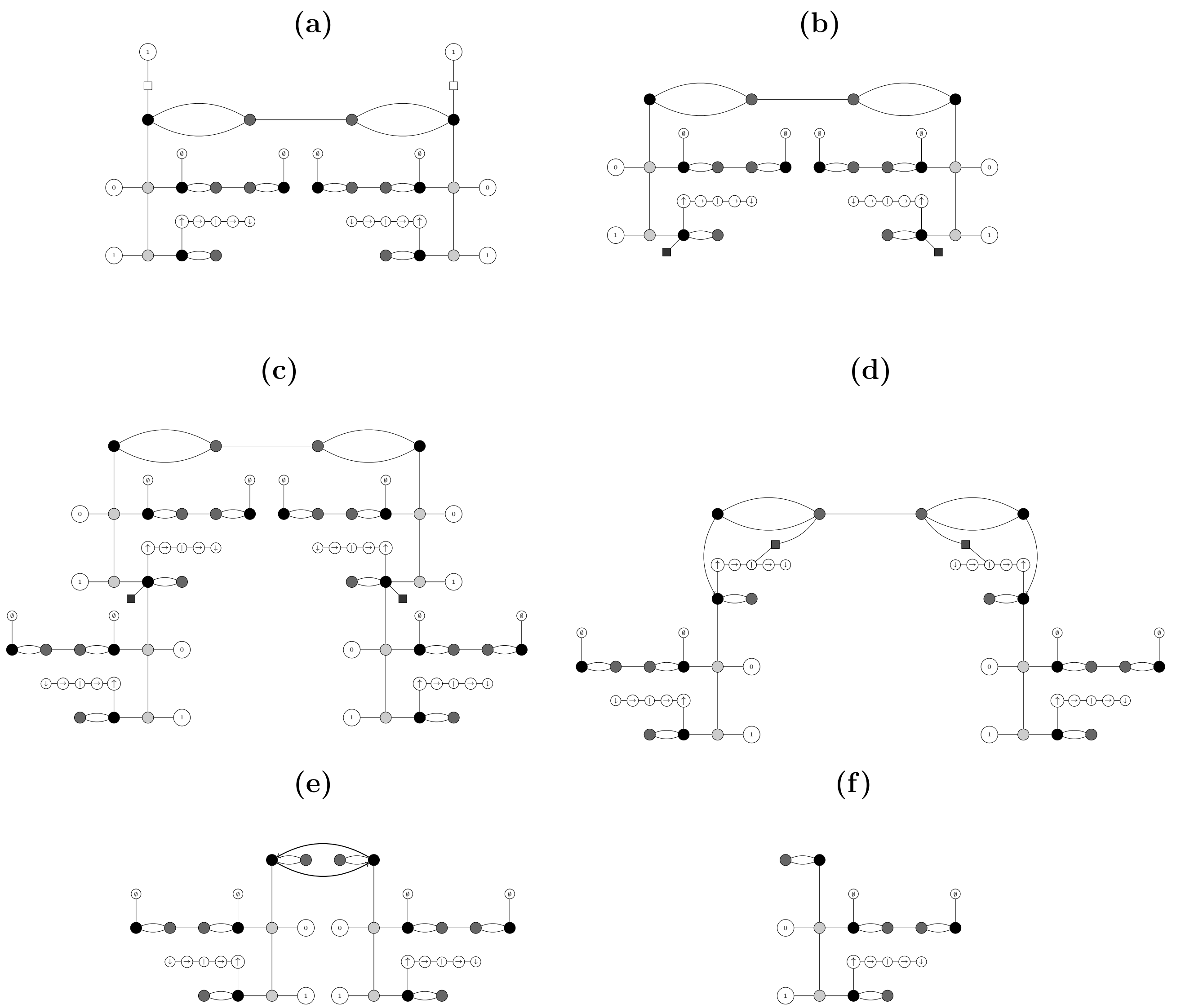

Figure 5.12: Steps of the simulation of the turtle local rule. (a) After the neighbourhood exploration. The two automata attached to each meta-vertices have detected the presence of another meta-vertex in the neighbourhood, and thus generated a vertex labelled 1. (b) The automaton travelled down the local rule and chose the second output subgraph. They will then start a DFS on the chosen subgraph. (c) the DFS detected a meta-vertex and decided to duplicate the local rule and attach a copy to it. (d) The DFS is over, the local rule is destroyed and the identification process is running. The automaton is on his way to reach the meta-vertex at the end of the address attached to the meta-vertex of the output subgrah. Two symbols of the address have been read: ↑ and →. (e) After reading the address the two meta-vertices are pointing toward each other and start the merging process. (f) After synchronization, the two meta-vertices and their local rule encodings are merged, and the simulation is over. To restart the simulation a new automaton can be attached to the meta-vertex.

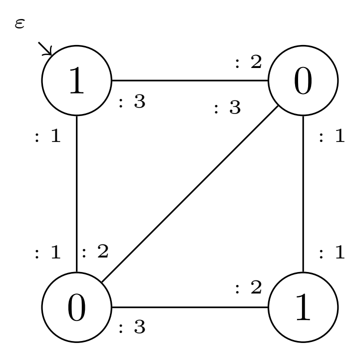

Figure 5.13: Generalized Cayley graph with set of port $\pi = \{1,2,3\}$ and with labels $\Sigma = \{0,1\}$. The incoming arrow on the top-left vertex indicates the pointed vertex ε. Its encoding is the string: $\$1;(1,1)\$0;(2,3)\$0(2,3)||;(1,1)\$1(2,3)||;$

Chapter 6

Causal dynamics of discrete geometrical spaces

Motivations. The motivation for developing this model of causal graph dynamics was to "free cellular automata off the grid", so as to be able to model any situation where a dynamics acts over a graph synchronously and locally, leading to changes in the topology, i.e. the notion of who is next to whom. When first introduced in [AD12], two examples were presented, both representing a distinct class of problems whose understanding might evolve through this model. The first example is that of a mobile phone network: mobile phones are modelled as vertices of the graph, in which they appear connected if they can call each other, i.e. one of them has the other as a contact. The second example is that of particles lying on a smooth surface and interacting with one another, but whose distribution influences the topology the smooth surface (cf. Heat diffusion in a dilating material, or even discretized General Relativity [Sor75]). Causal graph dynamics seems quite appropriate for modelling the first situation (or at least a stochastic version of it). Modelling the second situation, however, is not a short term perspective.

Are graphs enough? One of the main difficulties we face is that having generalized cellular automata to arbitrary graphs, we lost the capacity of interpreting our configurations as surfaces (for grids) or, more generally, as spaces of a given dimension. There exist, however, number of formalisms describing discrete (or "combinatorial") manifolds, that are very close to graphs (Abstract simplicial complexes, CW-complexes...). These work by considering a collection of "small" n-dimensional spaces and glueing them together, with some constraints, to form an approximation of a smooth manifold. The goal of this chapter is to study a correspondence between our model of graph and these objects, trying to define what could be causal transformations of discrete spaces. Section 6.1 focuses on the $2-$dimensional case, where we manage to give a definition of causal dynamics of discrete surfaces. Section 6.2 generalizes some of these results and states the next hurdles to the definition of causal dynamics of discrete spaces. To simplify the notation and focus on the interesting aspect of this work, we dropped the formalism of generalized Cayley graphs and went back to the formalism of [AD12]. All the results can be easily extended to generalized Cayley graphs and the definition of localizable dynamics introduced in chapter 3.

The content of this chapter is based on [AMW13], co-authored with Pablo Arrighi and Zizhu Wang.

6.1 The $2-$dimensional case

In the $2-$dimensional case, we want to define a correspondence between graphs of degree $|\pi| = 3$ and $2D$ CW complexes. In this formalism, a discrete surface is represented using triangles ($2D$ simplices, i.e. the simplest $2-$dimensional objects) glued along their facets. In this section, all graphs are of degree 3 with $\pi = \{a, b, c\}$.

6.1.1 Surfaces as Graphs

Correspondence. The straightforward way is to map each triangle to a vertex, and each facet of the triangle to an edge. The problem, then, is that we can no longer tell one facet from another, which leads to ambiguities (see Fig. 6.1 *Top row.*).

A first solution is to consider $2D$ *coloured simplicial complexes* instead. In these complexes, each of the three facets of a triangle has a different colour amongst $\{a, b, c\}$. Now each triangle is again mapped to a vertex, and each facet of the triangle to an edge, but this edge holds the colours of the facets it connects at its ends (see Fig. 6.1 *Bottom row.*).

The following provides a formal interpretation of those graphs into CW-complexes.

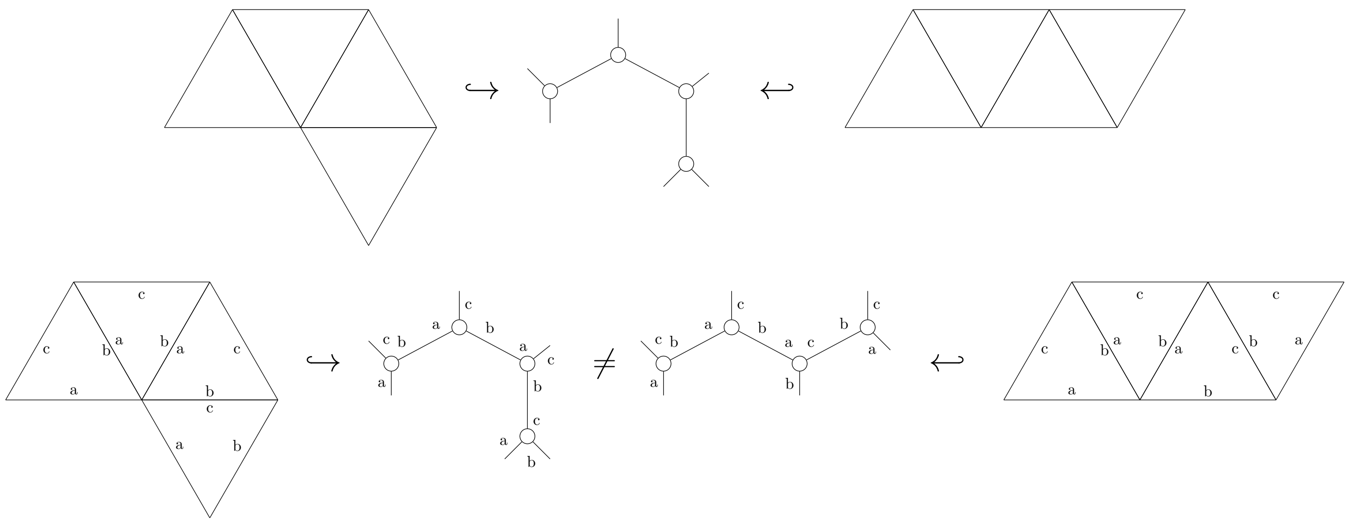

Figure 6.1: Complexes as graphs. *Top row.* The straightforward way to encode complexes as graphs is ambiguous. *Bottom row.* Encoding coloured complexes instead lifts the ambiguity. However, the fact that the extreme triangles share one point or not, is less obvious in the graph representation.

Definition 43 (Interpretation). *Given a graph G, its interpretation as a CW-complex $K(G)$ is such that:*

- *its set of triangles K_2 is $V(G)$.*
- *its set of segments K_1 is the quotient of $V(G) : \pi$ with respect to the equivalence: $u : p \equiv_1 v : q$ if and only if $\{u : p, u : q\} \in E(G)$. Elements of K_1 are denoted $u : \overline{p}$, to distinguish them from the following:*
- *its set of points K_0 is the quotient of $V(G) : \pi$ with respect to the equivalence: $u : p \equiv_0 v : q$ if and only if $\{u : (p{+}1), v : (q{-}1)\} \in E(G)$.*

A segment $u : \overline{p}$ has points $\{u : (p + o)$ modulo $\equiv_0 \mid o \in \{1, 2\}\,\}$.
A triangle u has segments $\{u : \overline{p}$ modulo $\equiv_1 \mid p \in \pi\,\}$. Notice that segments $u : \overline{p}$ and $u : \overline{q}$ have common point $u : \overline{p} \cap \overline{q}$.

This notion of coloured simplicial complex is not so common, however. It is more common to consider a version of coloured complexes where triangles can rotate freely, i.e. where we can permute the colours: a for b, b for c, c for

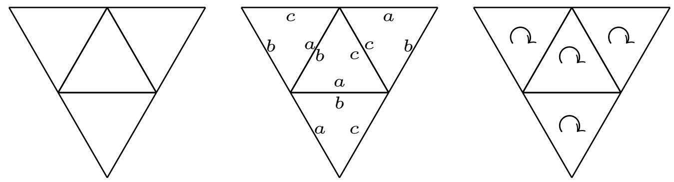

Figure 6.2: Complexes, Coloured complexes, Oriented Complexes

a, so that each triangle has a cyclic ordering of its facets but no privileged facet a. The cyclic ordering is then interpreted an orientation: when two facets are glued together in the complex, their orientation must be opposed, so that the two adjacent triangles have the same orientation. This leads to *oriented* $2D$ *simplicial complexes*. Fig. 6.2 summarizes the three kinds of $2D$ simplicial complexes we have mentioned. Definition 1 captured $2D$ coloured complexes as graphs. How can we capture oriented $2D$ simplicial complexes as graphs?

We need to define rotations of the vertices of the graphs in a way that corresponds to rotating the triangles of coloured complexes. Namely, vertex rotations simply permute the ports of the vertex, whilst preserving the rest of the graph:

Definition 44 (Vertex Rotation). *Let p_{ports} be some cyclic permutation over $\{a,b,c\}$, and p_{labels} be some bijection from Σ to itself such that $p^3_{\text{labels}} = id$. Let G be a graph and $u \in V(G)$ one of its vertices. Then $r_uG = G'$ is such that $V(G') = V(G)$ and:*

- $\forall v, w \neq u$ *we have* $\{v : i, w : j\} \in E(G) \wedge \Leftrightarrow \{v : i, w : j\} \in E(G')$.
- $\{u : i, v : j\} \in E(G) \Leftrightarrow \{u : p_{\text{ports}}(i), v : j\} \in E(G')$.
- $\sigma'(u) = p_{\text{labels}}(\sigma(u))$, *whereas* $\sigma'(v) = \sigma(v)$ *for* $v \neq u$.

(From now on in order to simplify notations we will drop all labels $\sigma(.) \in \Sigma$, though all the results of this chapter carry through to labelled graphs.) A rotation sequence $\overline{r}$ is a finite composition of rotations $r_{u_1}, r_{u_2}, \ldots$. Since rotations commute with each other, a rotation sequence can be seen as a multiset, i.e. a set whose elements can appear several times. Hence, the union of two rotation sequences $\overline{r_1} \sqcup \overline{r_2}$ refers to multiset union. Moreover, since $r_u^3 = Id$, we can consider that each rotation appears at most two times in a rotation sequence.

Second, we define the equivalence relation induced by the rotations. Using this equivalence relation, we can define graphs in which vertices have cyclic ordering of their edges, but no privileged edge a.

Definition 45 (Rotation Equivalence). *Two graphs G and H are rotation equivalent if there exists a sequence of rotations $\overline{r}$ such that $\overline{r}G = H$. This equivalence relation is denoted $G \equiv H$.*

Who is next to whom? On the one hand in the world of $2D$ simplicial complexes, two simplices are adjacent if they share a point. On the other hand in the world of graphs, two vertices are adjacent if they share an edge. These two notions do not coincide, as shown in Fig. 6.1. The figure also shows that two triangles share a point if and only if their corresponding vertices are related by a monotonous path:

Definition 46 (Alternating paths). *Let $\Pi = \{a, b, c\}^2$. We say that $u \in \Pi^*$ is a* path *of the graph G if and only if there is a sequence u of pairs of ports $q_i p_i$ such that it is possible to travel in the graph according to this sequence, i.e. there exists $v_0, \ldots, v_{|u|} \in V(G)$ such that for all $i \in \{0 \ldots |u| - 1\}$, one has $\{v_i : q_i, v_{i+1} : p_i\} \in E(G)$, with $u_i = q_i p_i$. We say that a path $u = q_0 p_0 \ldots q_{|u|} p_{|u|}$ alternates at $i = 0 \ldots |u| - 2$ if either $p_i = q_{i+1} + 1$ and $p_{i+1} = q_{i+2} - 1$, or $p_i = q_{i+1} - 1$ and $p_{i+1} = q_{i+2} + 1$. A path is k-*alternating *if and only if it has exactly k alternations. A path is* monotonous *if and only if it does not alternate.*

Thus distance one in complexes is characterized by the existence of a 0-alternating path. More generally, distance $k+1$ in complexes is characterized by the existence of a k-alternating path. Recall that our aim is to define a CA-like model of computation over these complexes. In CA models, each cell must have a bounded number of neighbours (or a bounded "star" in the vocabulary of complexes). This bounded-density of information hypothesis [Gan80] is the first justification for the following restriction upon the graphs we will consider:

Definition 47 (Bounded-star Graphs). *A graph G is* bounded-star *of bound s if and only if is monotonous paths are of length less or equal to s.*

Notice that the property is stable under rotation. A further justification for this restriction will be given later.

6.1.2 Causal dynamics of discrete Surfaces

This subsection formalizes causal dynamics of discrete Surfaces.
Rotation-commutating. First, we will restrict CGD so that they may use the information carried out by ports, but only as far as it defines an orientation. Formally, this means restricting to dynamics which commute with graphs rotations.

Definition 48 (Rotation-Commuting function). *A function F from $\mathcal{G}_\pi$ to $\mathcal{G}_\pi$ is rotation-commuting if and only if for all graph G and all sequence of rotations $\overline{r}$ there exists a sequence of rotations $\overline{r}^*$ such that $F(\overline{r}G) = \overline{r}^* F(G)$. Such an $\overline{r}^*$ is called a conjugate of $\overline{r}$. The definition extends naturally to functions from $\mathcal{D}_\pi$ to $\mathcal{G}_\pi$.*

Lemma 14 (Rotations factorisation). *For all finite set of graphs $G_1, ..., G_n$ and for all set of rotation sequences $\overline{r_1}, ..., \overline{r_n}$, if $G_1, ..., G_n$ are consistent with each other, and $\overline{r_1}G_1, ..., \overline{r_n}G_n$ are consistent with each other, then*

$$\bigcup_{i\in\{1,\dots,n\}} \overline{r_i}G_i = \left(\bigsqcup_{i\in\{1,\dots,n\}} \overline{r_i}\right) \bigcup_{i\in\{1,\dots,n\}} G_i$$

Proof. Notice that we only need to prove this result for the union of two graphs. Let us consider two graphs G_1, G_2 and two rotation sequences $\overline{r_1}, \overline{r_2}$ such that G_1 and G_2 are consistent and $\overline{r_1}G_1, \overline{r_2}G_2$ are consistent. Let us consider some rotation r_u appearing only once in $\overline{r_1}$. There are two possible cases:

- $r_u \notin \overline{r_2}$: In this case, r_u acts on $G_1 \setminus (G_1 \cap G_2)$. Indeed, if $u \in V(G_2)$ then $r_u G_1$ and G_2 can not be consistent as u has been rotated in the first graph and not in the second. As u only appears in a part of the graph that is left unchanged by the union, we have that $(r_u(\overline{r_1} \setminus r_u)G_1) \cup (\overline{r_2}G_2) = r_u[(\overline{r_1} \setminus r_u)G_1 \cup \overline{r_2}G_2]$.

- $r_u \in \overline{r_2}$: In this case, the two graphs $(\overline{r_1} \setminus r_u)G_1$ and $(\overline{r_2} \setminus r_u)G_2$ are consistent and the rotations sequences $(\overline{r_1} \setminus r_u)$ and $(\overline{r_2} \setminus r_u)$ leave the vertex u unchanged. It is easy to check that the graphs $r_u\left[(\overline{r_1} \setminus r_u)G_1 \cup (\overline{r_2} \setminus r_u)G_2\right]$ and $\overline{r_1}G_1 \cup \overline{r_2}G_2$ are the same.

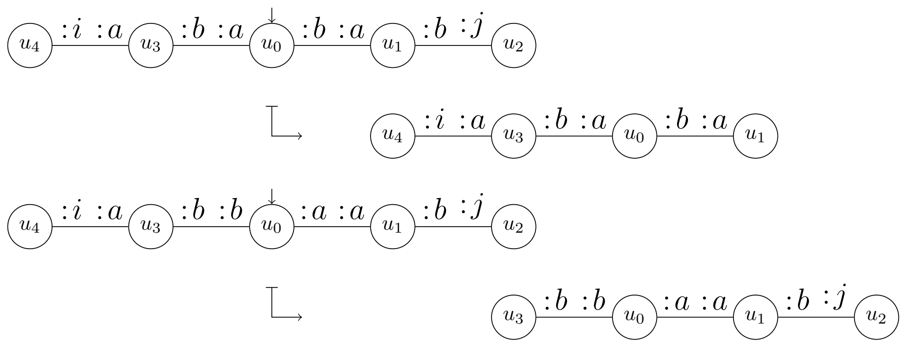

Figure 6.3: A non-rotation commuting local rule induces a rotation commuting CGD.

The case where r_u appears more than once in $\overline{r_1}$ can be proven similarly. By commuting all the rotations with the $\cup$ operator, we have that:

$$(\overline{r_1} \sqcup \overline{r_2})(G_1 \cup G_2) = (\overline{r_1}G_1) \cup (\overline{r_2}G_2)$$

□

The next question is "When is a CGD rotation-commuting?". More precisely, can we decide, given the local rule f of a CGD F, whether F is rotation-commuting? The difficulty is that being rotation-commuting is a property of the global function F. Indeed, a first guess would be that F is rotation-commuting if and only if f is rotation-commuting, but this turns out to be false.

Example 1 (Identity function). *Consider the local rule of radius* 1 *over graphs of degree* 2 *which acts as the identity in every cases but those given in Fig. 6.3. Because of these two cases, the local rule makes use the information carried out by the ports around the center of the neighbourhood. It is not rotation-commuting. Yet, the CGD it induces is just the identity, which is trivially rotation-commuting.*

Thus, unfortunately, some rotation-commuting F can be induced by a non-rotation-commuting f. Yet, fortunately, any rotation-commuting F can be induced by a rotation-commuting f.

Theorem 6 (Rotation commuting). *Let F be a localizable dynamics. F is rotation-commuting if and only if there exists a rotation-commuting local rule f which induces F.*

Proof. [$\Leftarrow$] Let us consider a rotation-commuting local rule f of radius r inducing a localizable dynamics F. Let G be a graph and u a vertex of G. The following sequence of equalities proves that F is rotation-commuting:

$$\begin{aligned} F(\overline{r}G) &= \bigcup_{v\in G} f(\overline{r}G_v^r) \\ &= \bigcup_{v\in G} \overline{r}_v^* f(G_v^r) \quad \text{(using } f \text{ rotation-commuting)} \end{aligned}$$

Using lemma 1, we can commute the union operator and the sequences of rotations $\overline{r}_v$ as follow:

$$F(\overline{r}G) = \left(\bigsqcup_{v\in G} \overline{r_v}^*\right) \bigcup_{v\in G} f(G_v^r) = \left(\bigsqcup_{v\in G} \overline{r_v}^*\right) F(G)$$

Less formally, we can commute rotations and unions by looking at the highest power with which the rotations appear at the right of the union operator.

[$\Rightarrow$] Let F be a rotation-commuting localizable function, and f a local rule inducing F. Informally, since $f(G_u^r)$ is included in $F(G)$ we know that as far as orientation is concerned f will indeed be rotation-commuting. However it may still happen that f, depending upon the orientation of G_u^r, will produce a smaller, or a larger, subgraph of $F(G)$. Therefore, we must define some $\tilde{f}$ which does not do that. Let us consider the following function $\tilde{f}$ from $\mathcal{D}_\pi$ to $\mathcal{G}_\pi$:

$$\forall G_u^r, \tilde{f}(G_u^r) = \bigcup_{\overline{r}} \overline{r}^{*-1} f(\overline{r}G_u^r)$$

with $\overline{r}^*$ a conjugate of $\overline{r}$ (given by F rotation-commuting).

- $\tilde{f}$ is well defined: By definition of $\overline{r}^*$ we have that:

$$\begin{aligned} & \forall \overline{r}, && f(\overline{r}G_u^r) \subset \overline{r}^* F(G) \\ \Rightarrow\ & \forall \overline{r}, && \overline{r}^{*-1} f(\overline{r}G_u^r) \subset F(G) \qquad (*) \\ \Rightarrow\ & \forall \overline{r_1}, \overline{r_2}, && \overline{r_1}^{*-1} f(\overline{r_1}G_u^r) \text{ and } \overline{r_2}^{*-1} f(\overline{r_2}G_u^r) \text{ consistent} \end{aligned}$$

- $\tilde{f}$ is a local rule: we can check that it inherits of the local rule properties of f.
- $\tilde{f}$ induces F:

$$\begin{aligned}
\bigcup_{v\in G}\tilde{f}(G_u^r) &= \bigcup_{v\in G}\left[\bigcup_{\overline{r}}\overline{r}^{*-1}f(\overline{r}G_u^r)\right]\\
&= \bigcup_{v\in G}\left[f(G_u^r)\cup\left(\bigcup_{\overline{r}\neq id}\overline{r}^{*-1}f(\overline{r}G_u^r)\right)\right]\\
&= F(G)\cup\bigcup_{v\in G}\left(\bigcup_{\overline{r}\neq id}\overline{r}^{*-1}f(\overline{r}G_u^r)\right)\\
&= F(G) \qquad \text{since } (*)
\end{aligned}$$

- $\tilde{f}$ is rotation-commuting: let us consider a sequence of rotations $\overline{s}$. We have:

$$\tilde{f}(\overline{s}G_u^r)=\bigcup_{\overline{r}}\overline{r}^{*-1}f(\overline{rs}G_u^r)$$

Let us define $\overline{t}=\overline{rs}$. As $\overline{r}$ spans all rotations sequences, $\overline{t}$ spans all rotations sequences. We can write:

$$\begin{aligned}
\tilde{f}(\overline{s}G_u^r) &= \bigcup_{\overline{t}}\overline{r}^{*-1}f(\overline{t}G_u^r)\\
&= \bigcup_{\overline{t}}\overline{r}^{*-1}\overline{t}^{*}\overline{t}^{*-1}f(\overline{t}G_u^r)\\
&= \left(\bigsqcup_{\overline{t}}\overline{r}^{*-1}\overline{t}^{*}\right)\bigcup_{\overline{t}}\overline{t}^{*-1}f(\overline{t}G_u^r) \quad \text{using lemma 1}\\
&= \left(\bigsqcup_{\overline{t}}\overline{r}^{*-1}\overline{t}^{*}\right)\tilde{f}(G_u^r)
\end{aligned}$$

□

Proposition 5 (Decidability of rotation commutation). *Given a local rule f, it is decidable whether f is rotation-commuting.*

Proof. There exists a simple algorithm to verify that f is rotation-commuting. Let r be the radius of f. We can check that for all disk $D\in\mathcal{D}_\pi^r$ and for

all vertex rotation r_u, $u \in V(D)$, we have the existence of a sequence $\overline{r}$ such that $f(r_u D) = \overline{r} f(D)$.
As the graph $f(D)$ is finite, there is finite number of sequences $\overline{r}$ to test. Indeed, if $|V(f(D))| = k$, we only have 3^k different sequences we can apply on $f(D)$ (for each vertex u, we can apply r_u 0,1 or 2 times). Notice that as f is a local rule, changing the names of the vertices in D will not change the structure of $f(D)$ and thus we only have to test the commutation property on a finite set of disks. □

Moreover, we have the more general result that:

Proposition 6 (Decidability of rotation commutation). *Given a local rule f, it is decidable whether f induces a rotation-commuting causal graph dynamics.*

Proof. Simply consider the construction of the local rule $\tilde{f}$ in the proof of Theorem 6. If we manage to build such a local rule, then, using Theorem 6, we will have that the induced CGD is rotation-commuting. On the other hand, if the construction fails, it will provide us a graph witness of the non-rotation-commutation of the induced dynamics. Moreover, the construction of $\tilde{f}$ is a finite process, hence the result.

□

Bounded-star preserving. Second, we will restrict CGD so that they preserve the property of a graph being bounded-star. Indeed, we have explained in Section 6.1.1 that the graph distance between two vertices does not correspond to the geometrical distance between the two triangles that they represent. By modelling CCD via CGD, we are guaranteeing that information does not propagate too fast with respect to the graph distance, but not with respect to the geometrical distance. The fact that the geometrical distance is less or equal to the graph distance is falsely reassuring: the discrepancy can still lead to an unwanted phenomenon as depicted in Fig. 6.4.
Of course we may choose not to care about geometrical distance. But if we do care, then we must make the assumption that graphs are bounded-star. This assumption will not only serve to enforce the bounded-density of information hypothesis. It will also relate the geometrical distance and the graph distance by a factor s. As a consequence, the guarantee that information does not propagate too fast with respect to the geometrical distance will be inherited from its counterpart in graph distance. In particular, it will forbid

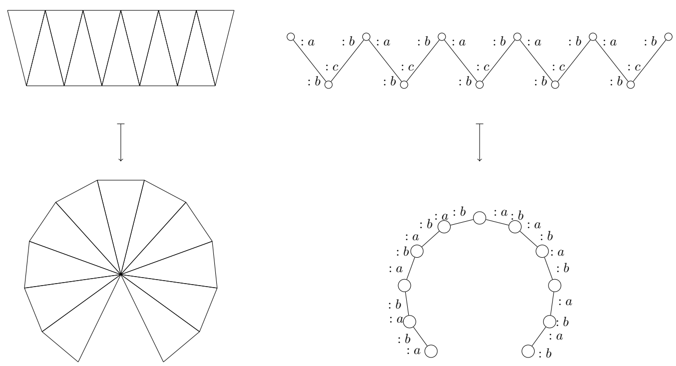

Figure 6.4: An unwanted evolution: sudden collapse in geometrical distance. *Left column:* in terms of complexes. *Right column.* In terms of graph representation.

the sudden collapse phenomenon of Fig. 6.4. All we need to do, then, is to impose that CCD take bounded-star graphs into bounded-star graphs. This can be decided from its local rule.

Definition 49 (Bounded-star preserving). *A CGD F is* bounded-star preserving *if and only if for all bounded-star graph G, $F(G)$ is also bounded-star. A local rule f is bounded-star preserving if and only if it induces bounded-star preserving a global dynamics F.*

Proposition 7 (Decidability of bounded-star preservation). *Given a local rule f and a bound s, it is decidable whether f is bounded-star preserving with bound s.*

Proof. We can, for each disk $D \in \mathcal{D}^r$ centred on a vertex u, consider a disk H of radius $2rs$ centered on u and containing D. Considering any 0-alternating path p of $f(D)$, the two following cases can appear:

- p is strictly contained in $f(D)$ and it can be checked whether its length is greater than s,
- p can be extended as a 0-alternating path in $f(H)$ by a length between 1 and s. In that case, we can also check if its length is strictly less than $s+1$.

By checking this property for each disk D of radius r and each H containing D, we can decide whether the image of a graph will contain 0-alternating path of length greater than s. □

This is our final definition of causal dynamics of discrete Surfaces:

Definition 50 (Causal dynamics of discrete Surfaces). *Let F be a causal graph dynamics over $\mathcal{X}_\pi$ with $|\pi| = 3$. F is a causal dynamics of discrete surface if F is rotation commuting and F is bounded star preserving.*

6.2 Toward an $n-$dimensional generalization

We now want to extend the previous results to $CW-$complexes of higher dimensions. As before, to each vertex of our graph will now correspond a small piece of $n-$dimensional space, and to each edge will correspond an identification of facets. The first difference between the $2-$ dimensional case and the general case lies in the fact that, when given two coloured triangles and two of their facets, there is only a single way of glueing them to obtain an orientable surface. This is not the case in higher dimension. For instance, when given two coloured tetrahedron and two of their facets, there is three different ways of glueing them while constructing an orientable $3-$dimensional space. In order to specify which of this glueing is used, we chose to use directed edges and to add a label on the edges describing the permutation used to transform the colors of the source vertex in the colors of the target vertex.

Definition 51 generalizes the correspondence between graphs of degree 3 and complexes of triangles to a correspondence between graphs of degree $n+1$ and complexes of $n-$simplices.

Definition 51 (Interpretation). *Given a graph G, its interpretation as a CW-complex $K(G)$ is such that:*

- *its set of n-simplices K_n is $V(G)$.*
- *For $0 \leq k \leq n-1$, its set of k-simplices K_k is the quotient of $V(G)\pi\mathcal{P}_{k+1}(\pi)$ with respect to the equivalence: $u\pi\{p_0, \dots p_k\} \equiv_k v\pi\{q_0, \dots q_k\}$ if and only if there exists $e = (u\pi p, r, v\pi q) \in E(G)$, such that $-r(p_0) = q_0, \dots, -r(p_k) = q_k$, and $p \neq p_0, \dots, p \neq p_k, q \neq q_0, \dots, q \neq q_k$.*

A $k+1$-simplex $u\pi\{p_0 \dots p_{k+1}\}$ has facets $u\pi\mathcal{P}_{k+1}(\{p_0 \dots p_{k+1}\})$ modulo $\equiv_k$. Notice that simplices $u\pi P$ and $u\pi Q$ have common simplex $u\pi P \cap Q$.

Figure 6.5 depicts a graph and its interpretation in the $3-$dimensional case.

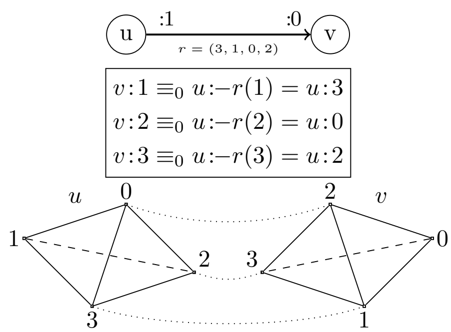

Figure 6.5: Interpreting graphs as CW-complexes: the different, rotated ways of glueing of two tetrahedrons along two given facets are specified on the edges.

We need to define rotations of the vertices of the graphs in a way that corresponds to rotating the $n-$simplices of a coloured complex while preserving their orientation. We also need to have a notion of vertex symmetry (i.e. vertex transformations that inverse its orientation). In the rest of this work, Π denotes the set of permutations over π.

Definition 52 (Vertex Rotation and vertex Symmetry). *Let G be a graph and $u \in V(G)$ one of its vertex and r an element of Π. Then $G' = (r@u)G$ is such that $V(G') = V(G)$ and:*

- $(u\pi p, s, v\pi q) \in E(G) \Leftrightarrow (u\pi p, s \circ r^{-1}, v\pi q) \in E(G') \wedge (v\pi q, r \circ s, u\pi p) \in E(G')$.
- $(v\pi i, s, w\pi j) \in E(G) \wedge v \neq u \wedge w \neq u \Leftrightarrow (v\pi i, s, w\pi j) \in E(G')$.

This transformation is called a rotation if r is an even permutation and a symmetry if it is odd (in this case it will be denoted by a s).
A rotation sequence $\overline{r}$ is a finite composition of rotations.
A symmetry sequence $\overline{s}$ is a finite composition of symmetries such that an odd number of symmetries is applied on each vertex of the graph.

Notice that, since the composition of two odds permutation is even, applying a symmetry to a strict subset of $V(G)$ produces an object which is not a graph (*ie* with even permutations on the edges). Moreover, applying two symmetries on the same vertex can be summed up as applying a rotation to the vertex. Therefore, when applying a sequence of symmetries, we must ensure that symmetries are applied an odd number of time on every vertex of the graph. We can also assume that a sequence of symmetries $\overline{s}$ applied to a graph G is in fact a sequence of $|V(G)|$ rotations $s_u, u \in V(G)$, as any composition of an odd number of symmetries is a symmetry. From now on, we will denote by $\overline{s}_u$ the symmetry applied to u in a symmetry sequence $\overline{s}$.

The definition of rotation equivalent graphs is left unchanged.

We can now try and define a notion of geometrical distance similar to definition 46.

Definition 53 (Paths). *We say that $e \in E(G)^*$ is a* path *of the graph G if and only if it is possible to travel in the graph according to this sequence, i.e. one has $e_i = (u_i\pi p_i, s_i, v_i\pi q_i) \in E(G)$, with $u_{i+1} = v_i$.*

Definition 54 generalizes the notion of $0-$alternating paths of section 6.1.

Definition 54 (Hinging paths). *Consider $e \in E(G)^*$ a* path *of the graph G with $e_i = (u_i\pi p_i, s_i, v_i\pi q_i) \in E(G)$, $u_{i+1} = v_i$. Fix $x_0 \in \pi$. We say that a path* hinges *around the point $u_0\pi x_0$ if and only if for all $i = 0 \dots |e| - 1$ we have $p_i \neq x_i$, with $x_{i+1} = s_i(x_i)$.*

We can now generalize the notion of alternating path:

Definition 55 (Alternating paths). *We say that a path k-*alternates *if $k+1$ is the minimal number of points $u_0\pi x_0, u_{i_1}\pi x_{i_1}, \dots u_{i_k}\pi x_{i_k}$ such that the subpaths $e_{0\dots i_1}, e_{i_1\dots i_2}, \dots e_{i_k\dots|e|}$ hinge around those points. A path is* monotonous *if and only if it 0-alternates.*

Thus distance one in complexes is characterized by the existence of a 0-alternating path. More generally, distance $k+1$ in complexes is characterized by the existence of a k-alternating path.

Definition 56 (Geometrical distance). *Two vertices u and v of a graph G are at geometrical distance k in G if there exists a $(k-1)$-alternating path between u and v in G and no j-alternating path with $j < k - 1$ between them. Given a single vertex u in G, the subgraph of neighbours of u in G is denoted* $\mathrm{Star}(G, u)$ *and corresponds to the subgraph induced by the vertices at geometrical distance 1 or less of u.*

Using this new definition of geometrical distance, we can easily generalizes the previous definition of bounded star graphs.

6.3 Hurdles toward a definition of causal dynamics of discrete geometrical spaces

Rotation commuting dynamics in higher dimension. While in the $2-$dimensional scope, the set of vertex rotations has a nice structure, this structure appears to be more complex in higher dimension. In particular two different rotations do not necessarily commute. It seems that the result of theorem 6 can be generalized, but the proof would require to keep track of the order of application of the vertex rotations.

Pseudo-manifolds and manifolds. Oriented $2-$dimensional complexes have a nice property: the underlying topological space is always smooth and without singularities. This is not the case in higher dimension. This property of having no singularities is a complex one. It is know to be undecidable in the general case, as it requires to be able to test the existence of an homeomorphism between a piece of our space and a $n-$dimensional ball. However, if we restrict ourselves to the set of bounded star graph of a given bound, it seems that we might be able to prove the decidability of this problem. This would require to define a proper notion of homeomorphism using local modifications of the graphs similar to Pachner moves [Pac91] .

Chapter 7

Conclusion

Summary of results

Generalized Cayley graphs. We introduced a new model of graphs generalizing the construction of Cayley graphs to arbitrary, bounded degree graphs. In these graphs, vertices are named relatively to a pointed vertex acting as the center of the graph. We defined several operations over generalized Cayley graphs, in particular a shift operation, consisting in moving the pointer on a different vertex, and a disk operation. We also proved that this set of graphs can be endowed with a compact metric.

Causal and localizable graph dynamics. We defined cellular automata over the set of generalized Cayley graphs in two fashions. First, we gave an axiomatic definition, relying on three properties, namely uniform continuity, shift-invariance and boundedness. Then, we gave a more constructive definition, relying on a notion of local rule applied synchronously and uniformly throughout the graph. We then proved that these two definitions are equivalent, hence generalizing Curtis-Hedlund-Lyndon theorem. We also proved that the composition of two causal graph dynamics is a causal graph dynamics, and that the application of a causal graph dynamics on a finite graph is a computable process.

Reversible causal graph dynamics. We proved that bijective instances of causal graph dynamics admit an inverse causal graph dynamics, hence generalizing a similar result of cellular automata theory. Moreover, we proved that these same instances of causal graph dynamics preserve the number of vertices of almost all finite graphs. Next, we proved that any reversible instance of causal graph dynamics can be implemented by a bounded-depth circuit of reversible local operations.

Intrinsic universality. We defined a notion of intrinsic simulation and intrinsic universality for causal graph dynamics. We began by tackling the simpler case of exhibiting an intrinsically universal set of local rules, namely the one having radius one and no labels. We then presented the construction of an intrinsically universal family of local rules, together with a universal constructing machine, able to construct any initial state of a simulation of the application of causal graph dynamics over a finite graph.

Causal dynamics of discrete geometrical space. We established a correspondence between our model of graphs and $\Delta-$complexes. In the $2-$dimensional scope, we characterize causal dynamics of discrete surfaces as causal graph dynamics acting on graphs of degree three having two properties: vertex rotation commutation and bounded star preservation. Moreover, we proved that it is equivalent for a causal graph dynamics over graphs of degree three to commute with vertex rotations or to be induced by a local rule which commutes with vertex rotations. We then stated the next steps toward the generalization of this characterization to higher dimensions.

Future works and next milestones

Quantum Causal Graph Dynamics. As discussed in chapter 4, the main purpose of the study of reversible causal graph dynamics, aside from the generalization of several classical cellular automata results, was to try and design a quantum version of the model. While the work described in section 4.3 would allow us to extend the block decomposition of section 4.2 to a broader set of Causal Graph Dynamics, it might come in handy to have a more constrained structure in our generalized block decomposition. A good lead would be to study a partitioned block decomposition of reversible causal graph dynamics, which is close to the block decomposition presented here, but is yet to achieve. This kind of decomposition has already been intensively studied in cellular automata theory, and led to some interesting universality results [MH89]. The idea is to implement a reversible cellular automaton using the compositions of two operations: a local exchange of information followed by an internal permutation. In the case of reversible Causal Graph

Dynamics, the decomposition might not be as simple, due to the dynamism of the neighbourhoods. The next two natural steps would be to, first, define a proper notion of partitioned reversible causal graph dynamics, and then, try and generalize to varying topologies the result of [ANW11] which states that causal unitary evolutions of a quantum graph automaton are localizable. Eventually, a good definition of quantum causal graph dynamics would lead to an interesting toy model for quantum gravitation.

Discrete general relativity. As stated at the beginning of this thesis, the model of causal graph dynamics is a good candidate to produce discrete toy models for general relativity. The long term goal of this approach would be to be able to simulate, using causal graph dynamics, Einstein's field equation over a discrete geometrical space. The work presented in chapter 6 is a first attempt toward being able to model causal dynamics of discrete manifolds, which would the first step to reach. After that, it seems natural to try and adapt all the key notions of Riemannian geometry, such as torsion and curvature to our model of discrete spaces, in the spirit of Regge calculus.

Bibliography

Personal bibliography

[AM12] P. Arrighi and S. Martiel. Generalized Cayley graphs and cellular automata over them. In *Proceedings of GCM 2012, Bremen, September 2012. Pre-print arXiv:1212.0027*, pages 129–143, 2012.

[AMN13] P. Arrighi, S. Martiel, and V. Nesme. Generalized Cayley graphs and cellular automata over them. *submitted (long version). Pre-print arXiv:1212.0027*, 2013.

[AMP15] P. Arrighi, S. Martiel, and S. Perdrix. Reversible Causal Graph Dynamics. arXiv pre-print, 2015.

[AMW13] P. Arrighi, S. Martiel, and Z. Wang. Causal dynamics over Discrete Surfaces. In *Proceedings of DCM'13, Buenos Aires, August 2013, EPTCS*, 2013.

[MM13] S. Martiel and B. Martin. Intrinsic universality of causal graph dynamics. In Turlough Neary and Matthew Cook, editors, Proceedings*, Machines, Computations and Universality 2013,* Zürich, Switzerland, 9/09/2013 - 11/09/2013, volume 128 of *Electronic Proceedings in Theoretical Computer Science*, pages 137–149. Open Publishing Association, 2013.

[MM15] S. Martiel and B. Martin. An intrinsically universal family of causal graph dynamics. *accepted at MCU 2015*, 2015.

[PBH+14] T. J. Proctor, K. E. Barr, B. Hanson, S. Martiel, V. Pavlović, A. Bullivant, and V. M. Kendon. Nonreversal and nonrepeating quantum walks. *Phys. Rev. A*, 89:042332, Apr 2014.

State of the art

[AD12] P. Arrighi and G. Dowek. Causal graph dynamics (long version). *Information & Computation journal, to appear. Preprint arXiv:1202.1098*, 2012.

[AFNT11] P. Arrighi, R. Fargetton, V. Nesme, and E. Thierry. Applying causality principles to the axiomatization of Probabilistic Cellular Automata. In *Proceedings of CiE 2011, Sofia, June 2011, LNCS*, volume 6735, pages 1–10, 2011.

[AN11] P. Arrighi and V. Nesme. A simple block representation of Reversible Cellular Automata with time-symmetry. In 17^{th} *International Workshop on Cellular Automata and Discrete Complex Systems, (AUTOMATA 2011), Santiago de Chile, November 2011.*, 2011.

[ANW08] P. Arrighi, V. Nesme, and R. F. Werner. Quantum cellular automata over finite, unbounded configurations. In *Proceedings of LATA, Lecture Notes in Computer Science*, volume 5196, pages 64–75. Springer, 2008.

[ANW10] P. Arrighi, V. Nesme, and R. Werner. Unitarity plus causality implies localizability. *J. of Computer and Systems Sciences*, 77:372–378, 2010. QIP 2010 (long talk).

[ANW11] P. Arrighi, V. Nesme, and R. Werner. Unitarity plus causality implies localizability (full version). *Journal of Computer and System Sciences*, 77(2):372–378, 2011.

[Arb88] M.A. Arbib. *The universal Turing machine*, chapter From universal Turing machines to self-reproduction, pages 177–189. Oxford Science Publications, 1988.

[BDG88] J.L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*. Springer Verlag, 1988.

[BFH87] P. Boehm, H.R. Fonio, and A. Habel. Amalgamation of graph transformations: a synchronization mechanism. *Journal of Computer and System Sciences*, 34(2-3):377–408, 1987.

[BT09] L. Boyer and G. Theyssier. On Local Symmetries and Universality in Cellular Automata. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009)*, Dagstuhl, Germany, 2009. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[CSC10] T. Ceccherini-Silberstein and M. Coornaert. *Cellular automata and groups*. Springer Verlag, 2010.

[CSS04] F. Ceccherini-Silberstein, T.; Fiorenzi and F. Scarabotti. The Garden of Eden Theorem for cellular automata and for symbolic dynamical systems. In *Random walks and geometry. Proceedings of a workshop at the Erwin Schrödinger Institute, Vienna, June 18 – July 13, 2001. In collaboration with Klaus Schmidt and Wolfgang Woess. Collected papers.*, pages 73–108. Berlin: de Gruyter, 2004.

[DFF+12] V. Danos, J. Feret, W. Fontana, R. Harmer, J. Hayman, J. Krivine, C. Thompson-Walsh, and G. Winskel. Graphs, rewriting and pathway reconstruction for rule-based models. In *FSTTCS 2012-IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 18, pages 276–288, 2012.

[DL01] J. O. Durand-Lose. Representing reversible cellular automata with reversible block cellular automata. *Discrete Mathematics and Theoretical Computer Science*, 145:154, 2001.

[DL04] V. Danos and C. Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69 – 110, 2004. Computational Systems Biology.

[DL09] J. O. Durand-Lose. Universality of Cellular Automata. In *Encyclopedia of Complexity and System Science*, pages 903–913. Springer, 2009.

[DMG08] B. Derbel, M. Mosbah, and S. Gruner. Mobile agents implementing local computations in graphs. *Graph Transformations*, pages 99–114, 2008.

[DS96] C. Dürr and M. Santha. A decision procedure for unitary linear quantum cellular automata. In *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, pages 38–45. IEEE, 1996.

[EL93] H. Ehrig and M. Lowe. Parallel and distributed derivations in the single-pushout approach. *Theoretical Computer Science*, 109(1-2):123–143, 1993.

[FAP90] V.V. Fedorchuk, A.V. Arkhangelskiui, and L.S. Pontriagin. *General topology I*, volume 1. Springer, 1990.

[Fre92] Edward Fredkin. Finite nature. *XXVIIth Rencotre de Moriond*, 1992.

[FT82] E. Fredkin and T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3):219–253, 1982.

[Gan80] R. Gandy. Church's thesis and principles for mechanisms. In *The Kleene Symposium*, Amsterdam, 1980. North-Holland Publishing Company.

[GMRT11] E Goles, P-E Meunier, Ivan Rapaport, and Guillaume Theyssier. Communication complexity and intrinsic universality in cellular automata. *Theoretical Computer Science*, 412(1):2–21, 2011.

[GR01] C. D. Godsil and G. Royle. *Algebraic graph theory*, volume 8. Springer-Verlag, New York, 2001.

[Gro99] M. Gromov. Endomorphisms of symbolic algebraic varieties. *Journal of the European Mathematical Society*, 1(2):109–197, April 1999.

[Gru10] S. Gruner. Mobile agent systems and cellular automata. *Autonomous Agents and Multi-Agent Systems*, 20:198–233, 2010. 10.1007/s10458-009-9090-0.

[GS08] J.L. Giavitto and A. Spicher. Topological rewriting and the geometrization of programming. *Physica D: Nonlinear Phenomena*, 237(9):1302–1314, 2008.

[Hed69] G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Math. Systems Theory*, 3:320–375, 1969.

[HM98] B. Hasslacher and D. A. Meyer. Modelling dynamical geometry with lattice gas automata. Expanded version of a talk presented at the Seventh International Conference on the Discrete Simulation of Fluids held at the University of Oxford, June 1998.

[Kar96] J. Kari. Representation of reversible cellular automata with block permutations. *Theory of Computing Systems*, 29(1):47–61, 1996.

[Kar11] J. Kari. Cellular Automata, Lecture notes. http:users.utu.fijkarica, 2011.

[KCN+10] A. Klales, D. Cianci, Z. Needell, D. A. Meyer, and P. J. Love. Lattice gas simulations of dynamical geometry in two dimensions. *Phys. Rev. E.*, 82(4):046705, Oct 2010.

[KK07] H.J. Kreowski and S. Kuske. Autonomous units and their semantics-the parallel case. *Recent Trends in Algebraic Development Techniques*, pages 56–73, 2007.

[KKBS05] W. Kurth, O. Kniemeyer, and G. Buck-Sorlin. Relational growth grammars–a graph rewriting approach to dynamical systems with a dynamical structure. *Unconventional Programming Paradigms*, pages 97–97, 2005.

[Löw93] M. Löwe. Algebraic approach to single-pushout graph transformation. *Theoretical Computer Science*, 109(1-2):181–224, 1993.

[Mar84] N. Margolus. Physics-like models of computation. *Physica D: Nonlinear Phenomena*, 10(1-2):81–95, 1984.

[Mar88] Norman H Margolus. Physics and computation. Technical report, DTIC Document, 1988.

[Mar07] Bruno Martin. Universal simulations by spatial machines. *Journal of Cellular Automata*, 2(3):203–217, 2007.

[Maz88] J. Mazoyer. An overview of the firing squad synchronization problem. In *Automata Networks*, pages 82–94. Springer, 1988.

[MH89] K. Morita and M. Harao. Computation universality of one-dimensional reversible (injective) cellular automata. *IEICE Trans. Inf. & Syst., E*, 72:758–762, 1989.

[Oll08] N. Ollinger. Universalities in cellular automata a (short) survey. In B. Durand, editor, *First Symposium on Cellular Automata "Journées Automates Cellulaires" (JAC 2008), Uzès, France, April 21-25, 2008. Proceedings*, pages 102–118. MCCME Publishing House, Moscow, 2008.

[Pac91] U. Pachner. Pl homeomorphic manifolds are equivalent by elementary 5hellingst. *European journal of Combinatorics*, 12(2):129–145, 1991.

[PR02] C. Papazian and E. Remila. Hyperbolic recognition by graph automata. In *Automata, languages and programming: 29th international colloquium, ICALP 2002, Málaga, Spain, July 8-13, 2002: proceedings*, volume 2380, page 330. Springer Verlag, 2002.

[RFH$^+$72] P. Rosenstiehl, J.R. Fiksel, A_ Holliger, et al. Intelligent graphs: Networks of finite automata capable of solving graph problems. *Graph theory and computing*, pages 219–265, 1972.

[Sor75] R. Sorkin. Time-evolution problem in Regge calculus. *Phys. Rev. D.*, 12(2):385–396, 1975.

[SW04] B. Schumacher and R. Werner. Reversible quantum cellular automata. arXiv pre-print quant-ph/0405174, 2004.

[Tae96] G. Taentzer. *Parallel and distributed graph transformation: Formal description and application to communication-based systems*. PhD thesis, Technische Universitat Berlin, 1996.

[Tae97] G. Taentzer. Parallel high-level replacement systems. *Theoretical computer science*, 186(1-2):43–81, 1997.

[TKM02] K. Tomita, H. Kurokawa, and S. Murata. Graph automata: natural expression of self-reproduction. *Physica D: Nonlinear Phenomena*, 171(4):197 – 210, 2002.

[TKM09] K. Tomita, H. Kurokawa, and S. Murata. Graph-rewriting automata as a natural extension of cellular automata. In Thilo Gross and Hiroki Sayama, editors, *Adaptive Networks*, volume 51 of *Understanding Complex Systems*, pages 291–309. Springer Berlin / Heidelberg, 2009.

[TMKK05] K. Tomita, S. Murata, A. Kamimura, and H. Kurokawa. Self-description for construction and execution in graph rewriting automata. *Advances in Artificial Life*, pages 705–715, 2005.

[VMPDJ11] S. Von Mammen, D. Phillips, T. Davison, and C. Jacob. A graph-based developmental swarm representation and algorithm. *Swarm Intelligence*, pages 1–12, 2011.

Appendix A

Adjacency sructures

In this appendix we provide an alternative, more algebraic, definition of generalized Cayley graphs, as a language endowed with a theory of equivalence. We then prove the equivalence between this definition and the definition given in chapter 2.

A.1 Paths structures

Definition 57 (Path structure). *Given a generalized Cayley graph X, we define the* structure of paths $S(X)$ *as the structure* $\langle L(X), \equiv_X \rangle$*. The* set of all path structures *is the set* $\{S(X) \mid X \in \mathcal{X}_\pi\}$*. It is written* $S(\mathcal{X}_\pi)$*.*

Given two generalized Cayley graphs, any difference between them shows up in their path structure.

Proposition 8 (Generalized Cayley graphs and path structures isomorphism). *The function $X \mapsto S(X)$ is a bijection between $\mathcal{X}_\pi$ and $S(\mathcal{X}_\pi)$.*

Proof. [Surjectivity]. By definition of $S(\mathcal{X}_\pi)$.
[Injectivity]. Let us suppose that $S(X) = S(Y)$, i.e. that $L(X) = L(Y)$ and $\equiv_X = \equiv_Y$. Then $\equiv_X$ and $\equiv_Y$ must have the same number of equivalence classes and so X and Y have the same number of vertices. Let us choose two graphs $P \in X$ and $Q \in Y$. For any vertex u of P, there is a unique equivalence class c of $\equiv_X$ such that the paths of c lead to u in P. Since $\equiv_X$ and $\equiv_Y$ are supposed equal, c is also an equivalence class of $\equiv_Y$. Conversely given c an equivalence class of $\equiv_Y$, there is a unique v of Q such that the

paths of c lead to v in Q. Then, the paths which point to u in P are the same as those which point to v in Q. We can now define a function R which maps each vertex u in P to its corresponding vertex v in Q. Because this is a bijection, we can then extend R to be a bijection over the entire set V. Let us consider two vertices u and u' in P linked by and edge $\{u : i, u' : j\}$ and their corresponding vertices v and v' in Q. As $P \in X$, we have that the equivalence classes $\tilde{u}.ij = \tilde{u'}$. As the classes representing v and v' are equal to $\tilde{u}.ij$ and $\tilde{u'}$. Thus R is a graph isomorphism, and P and Q are isomorphic. This is true for every $P \in X$ and $Q \in Y$ thus $X = Y$. $\square$ $\square$

A.2 Paths as languages

Inversely, we could have started by defining a certain class of languages endowed with an equivalence, namely adjacency structures, and then asked whether the path structures of generalized Cayley graphs fall into this class. This is the purpose of the following definitions and lemma.

Definition 58 (Completeness). *Let $L \subseteq \Pi^*$ be a language and $\equiv_L$ an equivalence on this language. The tuple $(L, \equiv_L)$ is said to be* complete *if and only if*

(i) $\forall u, v \in \Pi^* \quad u.v \in L \Rightarrow u \in L$

(ii) $\forall u, u' \in L \, \forall v \in \Pi^* \quad (u \equiv_L u' \wedge u.v \in L) \Rightarrow (u'.v \in L \wedge u'.v \equiv_L u.v)$

(iii) $\forall u \in L \, \forall a, b \in \pi \quad u.ab \in L \Rightarrow (u.ab.ba \in L \wedge u.ab.ba \equiv_L u)$

The completeness conditions aim at making sure that $(L, \equiv_L)$, seen as some algebra of paths, is complete. Indeed: (i) means that "a shortened path remains a path"; (ii) means that "Equivalent paths from A to B admit the same prolongations from B to C which lead to equivalent paths from A to C"; (iii) means that "if a step takes you from A to B, the inverse step takes you from B to A".

Definition 59 (Adjacency structure). *Let $L \subseteq \Pi^*$ be a language and $\equiv_L$ an equivalence on this language. The tuple $(L, \equiv_L)$ defines an* adjacency structure *if and only if it is complete and*

$$\forall u, u' \in L \, \forall a, b, c \in \pi \quad (u \equiv_L u' \wedge u.ab \in L \wedge u'.ac \in L) \Rightarrow b = c.$$

When this is the case, L is referred to as an adjacency language *and* $\equiv_L$ *as an* adjacency equivalence. *We denote by* $\langle L, \equiv \rangle$ *an adjacency structure of langage L and equivalence relation* $\equiv$*. The* set of all adjacency structures *is written* $\mathcal{S}_\pi$*. From now on, S will represent an element of* $\mathcal{S}_\pi$*.*

The added adjacency structure condition aims at making sure that $(L, \equiv_L)$, seen as some algebra of paths, is port-unambiguous, meaning that "once at some place A, taking port a leads to a definite place B".

Definition 60 (Associated (generalized Cayley) graph). *Let S be some adjacency structure* $\langle L, \equiv_L \rangle$*. Let $P(S)$ be the pointed graph* $(G(S), \tilde{\varepsilon})$*, with $G(S)$ such that:*

- *The set of vertices $V(G(X))$ is the set of equivalence classes of S;*
- *The edge $\{\tilde{u} : a, \tilde{v} : b\}$ is in $E(G(S))$ if and only if $u.ab \in L$ and $u.ab \equiv_L v$, for all $u \in \tilde{u}$ and $v \in \tilde{v}$.*

We define the associated graph *to be $G(S)$. We define the* associated pointed graph *to be $P(S)$. We define the* associated generalized Cayley graph *to be $X(S)$.*

Soundness: The properties of adjacency structures ensure that the ports of the vertices are not used several times. Moreover, $G(S)$ (and thus $P(S)$) are connected as every vertex is path connected to the vertex $\tilde{\varepsilon}$.

Lemma 15 (Path structures are adjacency structures). *Let X be a generalized Cayley graph. Then $S(X)$ is an adjacency structure. Hence* $S(\mathcal{X}_\pi) \subseteq \mathcal{S}_\pi$*.*

Proof. [Completeness]. If $u.v$ is a valid path in X, then the truncated path u is a valid path in X and belongs to $L(X)$.
If two paths u and v in X lead to the same vertex, i.e. $u \equiv_X v$, then extending u and v by the same path w will still lead to the same vertex i.e. if $u.w \in L(X)$ $u.w \equiv_X v.w$.
If $u.ab$ is a valid path in X then the extension $u.ab.ba$ consisting in going back on the last visited vertex is still a valid path and leads to the vertex pointed by u.
Summarizing, the completeness properties are verified by construction of the language of path $L(X)$ and the relation $\equiv_X$.

[Adjacency structure]. Let us consider two paths u and v in $L(X)$ and three ports a, b, c such that $u \equiv_X v$ and $u.ab \equiv_X u.ac$. Then, for the graph X to be well defined we have that $b = c$. $\square$ $\square$

Not only do we have that path structures are adjacency structures, but it also turns out that any adjacency structure can be generated this way, i.e. it is the path structure of some generalized Cayley graph.

Proposition 9 (Adjacency structures are path structures). *Let S be some adjacency structure. The equality $S = S(X(S))$ holds. Hence $\mathcal{S}_\pi = S(\mathcal{X}_\pi)$.*

Proof. Let $S = \langle L, \equiv_L \rangle$ and $S' = S(\tilde{P}(X)) = \langle L', \equiv_{L'} \rangle$. Next, we will write $S \subseteq S'$ if and only if $L \subseteq L'$ and $\equiv_L \subseteq \equiv_{L'}$, with the relations $\equiv_L$, $\equiv_{L'}$ viewed as subsets of $(L \cup L')^2$.

[$S \subseteq S(X(S))$]:

Let us consider $w \in L$. By construction of $X(S)$, there exists a path w in $X(S)$. By definition of the function X, we have that this path will be represented by the word $w \in L'$. Now, let us consider two words u and v in L such that $u \equiv v$. By construction of $X(S)$, u and v will be two paths of $X(S)$ leading to the same vertex. By definition of the function X, the two words u and v in L' will be equivalent regarding to the relation $\equiv'$.

[$S(X(S)) \subseteq S$]:

Let $w' \in L'$. By definition there exists a path ω' in $X(S)$ labeled by w' from the pointed vertex to a vertex u. By Definition 59 there exists a word in L describing the path ω', hence $w' \in L$. Similarly we prove the inclusion $\equiv_{L'} \subseteq \equiv_L$.

$\square$

A.3 Graphs as languages

Generalized Cayley graphs. Summarizing, $S(.)$ is bijective from Proposition 8 and $S \circ X = Id$ from Proposition 9, thus X is bijective, i.e. the following theorem comes out as a corollary:

Theorem 7 (Generalized Cayley graphs and adjacency structures isomorphism). *The function $X \mapsto S(X)$ is a bijection between $\mathcal{X}_\pi$ and $\mathcal{S}_\pi$,* *whose inverse is the function* $S \mapsto X(S)$.

Therefore, $\mathcal{X}_\Sigma$ and $\mathcal{S}_\Sigma$ are the same set, namely the set of *generalized Cayley graphs*.
Discussion. Generalized Cayley graphs extend Cayley graphs:

Proposition 10 (Recovering Cayley graph). *Consider H a group with law $*$ and generators the finite set $h = \{a, b, \ldots\}$. Let $\pi = \{a, a^{-1} \mid a \in h\}$ be the generators together with their inverses, $\overline{\pi} = \{(a, a^{-1}), \mid a \in \pi\}$ the generators paired up with their inverses. We define L to be $\overline{\pi}^*$. Consider the morphism mapping:*

- *a in π to $\overline{a} = (a, a^{-1})$ in $\overline{\pi}$*
- *the term $a * v$ in H to $\overline{a}.\overline{v}$ in L*
- *the equivalence $u = v$ over H to the equivalence $\overline{u} \equiv_L \overline{v}$ over L.*

Then, $S = \langle L, \equiv_L \rangle$ is an adjacency structure, and the generalized Cayley graph X coincides with the Cayley graph of H.

Proof. All of the adjacency structures conditions are met:

(i) $\overline{u}.\overline{v} \in L \Rightarrow \overline{u} \in L$ by definition of L.

(ii) $\overline{u} \equiv_L \overline{u}' \Rightarrow \overline{u}'.\overline{v} \equiv_L \overline{u}.\overline{v}$, since $u = u' \Rightarrow u * v = u * v'$.

(iii) $\overline{u}.\overline{a} \Rightarrow \overline{u}.\overline{a}.\overline{a^{-1}} \equiv_L \overline{u}$, since $u * a * a^{-1} = u$.

(-) $(\overline{u} \equiv_L \overline{u}' \wedge \overline{u}.(a, b) \in L \wedge \overline{u}'.(a, c) \in L) \Rightarrow b = c = a^{-1}$ by definition of L.

□

One might have thought that any adjacency structure over the language $\langle L, \equiv_L \rangle$, with $L = \overline{\pi}^*$ is a Cayley graph, but this is not the case: the fact that $\equiv_L$ corresponds to group equality does matter in the above proposition. The Petersen graph, for instance, can be endowed with such an adjacency structure, while being famously not a Cayley graph [GR01], see Figure A.3.
But generalized Cayley graphs extend Cayley graphs in a much wider way than just including Petersen-like graphs. Indeed, whereas Cayley graphs are highly symmetric, generalized Cayley graphs can be *arbitrary connected*

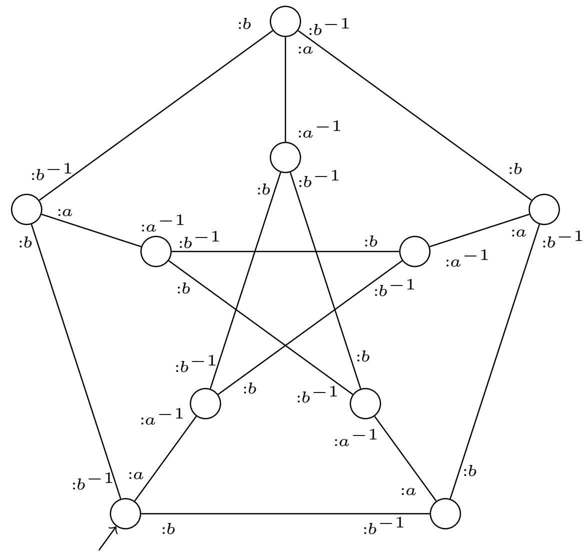

Figure A.1: *The Petersen graph as a generalized Cayley graph structure.*

graphs of bounded degree. Still, this extension is an advantageous one, since all of the key features of Cayley graphs remain: We are able to name vertices relative to a point, through the word describing the path from that point, and in fact the topology of the graph describes the equivalence structure upon words. We have a well-defined notion of translation, which is described as part of the basic operations upon these graphs in Section 2.3. We can define a distance between theses graphs, which makes $\mathcal{X}_{\Sigma,\Delta,\pi}$ a compact metric space, as done in Section 2.4.