Math. Struct. in Comp. Science (2018), *vol.* 28, *pp.* 340–383. © Cambridge University Press 2017
doi:10.1017/S0960129517000044 First published online 29 May 2017

Cellular automata over generalized Cayley graphs

PABLO ARRIGHI[†], SIMON MARTIEL[‡] and VINCENT NESME[§]

[†]*Aix-Marseille Univ., CNRS, LIF, Marseille and IXXI, Lyon, France*
Email: pablo.arrighi@univ-amu.fr
[‡]*INRIA Saclay, ENS Cachan, LSV, 61 avenue du président Wilson 94235 Cachan*
Email: martiel@lsv.ens-cachan.fr
[§]*Université de Grenoble, LIG, 220 rue de la chimie, 38400 Saint-Martin-d'Hères, France*
Email: vnesme@imag.fr

Received 17 June 2015; revised 27 December 2016

It is well-known that cellular automata can be characterized as the set of translation-invariant continuous functions over a compact metric space; this point of view makes it easy to extend their definition from grids to Cayley graphs. Cayley graphs have a number of useful features: the ability to graphically represent finitely generated group elements and their relations; to name all vertices relative to an origin; and the fact that they have a well-defined notion of translation. We propose a notion of graphs, which preserves or generalizes these features. Whereas Cayley graphs are very regular, generalized Cayley graphs are arbitrary, although of a bounded degree. We extend cellular automata theory to these arbitrary, bounded degree, time-varying graphs. The obtained notion of cellular automata is stable under composition and under inversion.

1. Introduction

1.1. *Synopsis of the model*

Here is a brief and informal overview of the topic.

Cellular Automata: from grids to graphs. Cellular automata (CA) consist of a $\mathbb{Z}^n$ grid of identical cells, each of which may take a state among a finite set Σ. Thus, the configurations are in $\Sigma^{\mathbb{Z}^n}$. These evolve by the next state of a cell from the state of its neighbours according to a fixed local rule f, which is applied synchronously and homogeneously across space. CA constitute the most established model of computation that accounts for euclidean space. They were introduced to describe self-replicating machines, but are used to model synchronization problems as well as an immense variety of physical, biological, sociological phenomena. This paper deals with a two-fold extension of CA. First, the underlying grid is extended to an arbitrary bounded degree graph G. Informally, this means having configurations in $\Sigma^{V(G)}$. Second, the graph itself is allowed to evolve over time. Informally, this means having configurations in $\bigcup_G \Sigma^{V(G)}$. This leads to a model where a local rule f is applied synchronously and homogeneously on every possible subdisk of the input graph, thereby producing small patches, whose union constitute the output graph. Figure 1 illustrates the concept of these CA over graphs.

Why this extension?. There are countless situations in which some agents (e.g. physical systems, computer processes, biochemical agents…) interact with their neighbours,

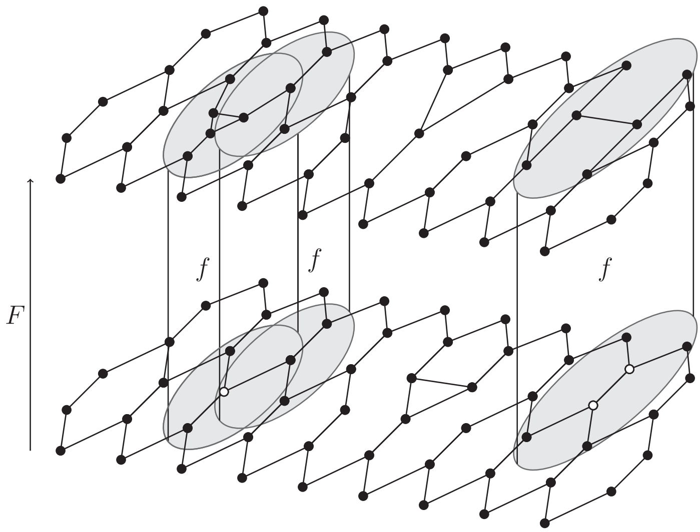


Fig. 1. Representation of a graph transformation F induced by the homogeneous and synchronous application of a local rule f on every vertex of a graph.

leading to a global dynamics, the state of each agents evolving through the interactions. In most of these situations, the notion of who is next to whom also varies in time (e.g. agents become physically connected, get to exchange contact details, move around. . .). For a very concrete example, think of a mobile phone network: the agents are the mobile phones, and their neighbours the mobile phones that they can call, i.e. the ones whose phone numbers they have. The entire graph evolves in time, but this global dynamics emerges from a neighbour-to-neighbour interactions. Ideally, it does so in a causal manner: new contacts are always contacts of contacts. Sometimes new mobile phones get created, and others get thrown out. For a more theoretical example, think of discrete versions of relativity à la Regge calculus (Regge 1961; Sorkin 1975): the agents are the simplicial complexes describing the spacelike geometry, and their neighbours are those which are adjacent. The entire graph evolves in time, but in a way which respects causality: information cannot propagate faster than light. At the intuitive level, the general concept of a global dynamics caused by neighbour-to-neighbour interactions and with a time-varying neighbourhood, is therefore not so difficult to visualize. This paper deals with the provision of a theoretical framework, in which such phenomena may eventually be modelled.

Constructive versus Axiomatic approach. There are two main approaches to define CA. The one with the local rule f is the constructive one, as one immediately sees how to compute the evolution of a CA, and even how to enumerate all possible CA. But CA can also be defined in a more topological way as being exactly the shift-invariant continuous functions from $\Sigma^{\mathbb{Z}^n}$ to itself, with respect to a certain metric. Through a compactness argument, the two approaches are equivalent. This topological approach is difficult to carry through to CA over graphs. In Arrighi and Dowek (2012), one of the authors, together with Dowek, proposed a first formalism for an extension of CA to time-varying

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

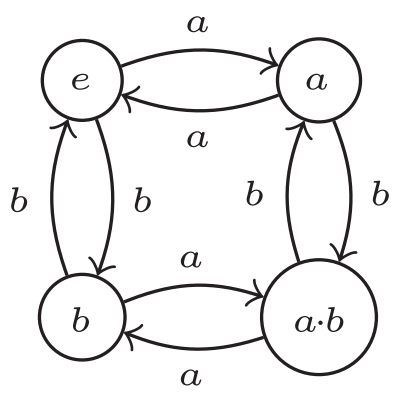


Fig. 2. A Cayley graph representation of the group with two generators $\{a, b\}$ with relations $a \cdot b = b \cdot a$ and $a = a^{-1}$ and $b = b^{-1}$. This group has four elements (e, a, b and $a \cdot b$), hence the presence of four vertices in its representation.

graphs, without achieving a proper generalization of the classical topological definition of CA. In particular, they failed to provide a set of configurations (i.e. labelled graphs) that could be equipped with a compact topological structure. The main contribution of this paper is the construction of such a set of configurations and the formalization of a topological definition for CA over time-varying graphs. The construction was achieved through the definition of a new set of graphs, the set of pointed graphs modulo – which we refer to as generalized Cayley graphs. It is worth the trouble, as the equivalence between the axiomatic and the constructive definitions of CA over generalized Cayley graphs shows the generality, the composability, and under certain conditions also the reversibility of CA over graphs.

1.2. *Generalizing Cayley graphs*

Cayley graphs are graphs associated to a finitely generated group, more precisely to a finite set of generators and their inverses. For instance, let this set be $\pi = \{a, a^{-1}, b, b^{-1}, \ldots\}$. Then the vertices of the graph can be designated by words on π, e.g. $a, a^2, a^{-1}, a \cdot b, \ldots$, but more precisely they are the equivalence classes of these words with respect to the group equality, e.g. $b^{-1} \cdot b \cdot a$ and a designate the same vertex, ab and ba designate the same vertex for a commutative group, etc. (often this group equality is specified in terms of finite set of relations). Figure 2 represents a Cayley graph of a commutative group with 2 generators. The edges are those pairs $(u, u.a)$. The same group can admit non-isomorphic Cayley graphs, depending on the choice of generators. Cayley graphs have been used intensively because they have a number of useful features:

- Once an origin has been chosen, all other vertices can be named relatively to the origin.
- The resulting graph represents the group, i.e. the set of elements and their equality.

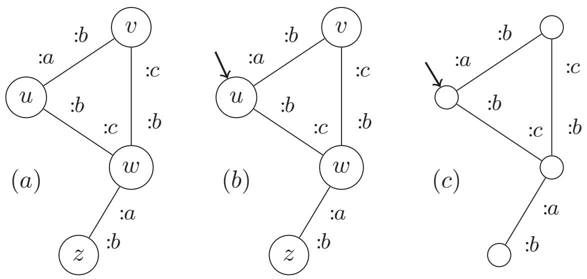


Fig. 3. *The different types of graphs.* (a) A graph. (b) A pointed graph. (c) A generalized Cayley graph. In (c), vertices have no name and the formal way of describing this graph structure is given in Section 7.

- There is a well-defined notion of translation of the graph, which corresponds to changing the point representing the origin, or equivalently applying an element to all vertices.
- The set of configurations over a given Cayley graph (i.e. labellings of the graph) can be endowed with a structure of a compact metric space, which has been used in order to define CA over them.

Yet, Cayley graphs are very regular. We propose a notion of graph which conserves or generalizes all of these features to arbitrary, bounded-degree graphs. Basically, generalized Cayley graphs are your usual connected, undirected, possibly infinite, bounded-degree graphs, but with a few added twists:

- Each vertex has *ports* in a finite set π, similarly to the graph model used in Kappa models, for instance (Danos et al. 2012). 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.
- The graphs are rooted i.e. 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 from 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 generalized Cayley graphs (see Figure 3c) of ports π, vertex labels Σ and edge labels Δ are denoted $\mathcal{X}_{\Sigma,\Delta,\pi}$. Thorough definitions for it are provided in Section 2. Yet, Figure 3 summarizes their construction as pointed graphs whose vertex names are dropped, and is in fact sufficient to follow most of the paper.

It is clear from this definition that there are generalized Cayley graphs which are not Cayley graphs. Cayley graphs, on the other hand, can be recovered as shown in Section 7.

1.3. *Generalizing cellular automata*

CA can also be characterized in purely mathematical terms as the set of translation-invariant continuous functions (Hedlund 1969) for a certain compact metric. As a consequence, CA definitions are quite naturally extended from grids to Cayley graphs, where most of the theory carries through Róka (1999) and Ceccherini-Silberstein and Coornaert (2010). In this case, the set of configuration is Σ^G, where G is a fixed Cayley graph. Moving on, there have been several approaches to generalize CA not just to Cayley graphs, but to arbitrary connected graphs of bounded degree:

- With a fixed topology, in order to describe certain distributed algorithms (Derbel et al. 2008; Gruner 2010; Papazian and Remila 2002), or to generalize the Garden-of-Eden theorem (Ceccherini-Silberstein et al. 2004; Gromov 1999).
- Through the simulation environments of Giavitto and Spicher (2008), Von Mamme et al. (2010) and Kurth et al. (2005) which offer the possibility of applying a local rewriting rule simultaneously in different non-conflicting places.

Whilst the above-cited works extend the CA from grids to graphs, the graph remains untouched by the evolution. I.e. configurations are in Σ^G for some given fixed graph, and the evolution is a function over this space. CA over time-varying graphs appear

- through concrete instances advocating the concept of CA extended to time-varying graphs as in Tomita et al. (2009), Kreowski and Kuske (2007), Klales et al. (2010), some of which are advanced algorithmic constructions (Tomita et al. 2002, 2005);
- through Amalgamated Graph Transformations (Boehm et al. 1987; Löwe 1993; Métivier and Sopena 1997) and Parallel Graph Transformations (Ehrig and Lowe 1993; Kreowski and Kuske 2011; Ryszka et al. 2015a,b; Taentzer 1996, 1997), which work out rigorous ways to apply a local rewriting rule synchronously throughout a graph.

In the above-cited papers, and in the present one, there is a set of configurations of all possible graphs (informally, $\bigcup \Sigma_i^G$), and the evolution is a self map over this set, thus allowing for time-varying graphs. Yet, the approach of this paper is different in the sense that it generalizes Cayley graphs, and then applies the mathematical characterization of CA as the set of translation-invariant continuous functions in order to generalize CA. Compared with the above mentioned CA papers, the contribution is to extend the fundamental structure theorems about CA to arbitrary, connected, bounded degree, time-varying graphs: A Curtis–Hedlund–Lyndon like theorem is proven together with its consequences on the composability and invertibility of CA over generalized Cayley 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.

Causal Graph Dynamics. The work (Arrighi and Dowek 2012) by Dowek and one of the authors already achieves an extension of CA to arbitrary, bounded degree, time-varying graphs, also through a notion of continuity, with the same motivations. However, graphs in Arrighi and Dowek (2012) lack a compact metric, which is left as an open question. As a consequence, all the necessary facts about the topology of Cayley graphs get reproven. It also leaves open whether causal graph dynamics are computable. These issues vanish in the new formalism which suggests that the new formalism itself is the main contribution of this paper.

1.4. *This paper*

Section 2 provides a generalization of Cayley graphs, as labelled pointed graphs modulo. Section 3 provides the basic operations upon generalized Cayley graphs. Section 4 provides facts about the topology of generalized Cayley graphs. It follows that continuous functions are uniformly continuous. Section 5 establishes a notion of CA over generalized Cayley graphs. An equivalence theorem between an existential and a constructive approach is given. It also shows that recognizing valid local rules is a recursive task, as well as that of computing their effect over finite graphs; this grants our model the status of a model of computation. Section 6 provides important corollaries: the stability of the notion of CA over generalized Cayley graphs under composability and taking the inverse function. It quickly mentions the status of the Garden of Eden theorem in this setting, as well as interesting subclasses of graph dynamics. Section 7 provides a characterization of generalized Cayley graphs, as a certain kind of algebraic structure on words.

2. Generalized Cayley graphs

The present section formalizes generalized Cayley graphs. Some readers may be content with their informal description of Subsection 1.2. Others may wish to deepen their fundamental algebraic properties, in terms of languages and relations, with due comparison with Cayley graphs: this is done in Section 7.

Vertex names. Let π be a finite set, $\Pi = \pi^2$ be its square 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 are up to isomorphism, we still need to manipulate plain graphs, non-modulo, at different stages. The vertices of these graphs non-modulo (See Figure 3a) are uniquely identified by a name u in V. (This particular choice of the universe of names is actually irrelevant until Definition 2.8, when it becomes natural.)

Construction. Definitions 2.1 to 2.4 are as in Arrighi and Dowek (2012). The first two are reminiscent of the many papers seeking to generalize CA to arbitrary, bounded degree, fixed graphs (Boehm et al. 1987; Ceccherini-Silberstein et al. 2004; Derbel et al. 2008; Ehrig and Lowe 1993; Gromov 1999; Gruner 2010; Kreowski and Kuske 2007; Löwe 1993; Papazian and Remila 2002; Taentzer 1996, 1997; Tomita et al. 2002, 2009, 2005). They are illustrated by Figure 3a.

Definition 2.1 (Graph). A *graph* G is given by

- an at most countable subset $V(G)$ of V, whose elements are called *vertices*;

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

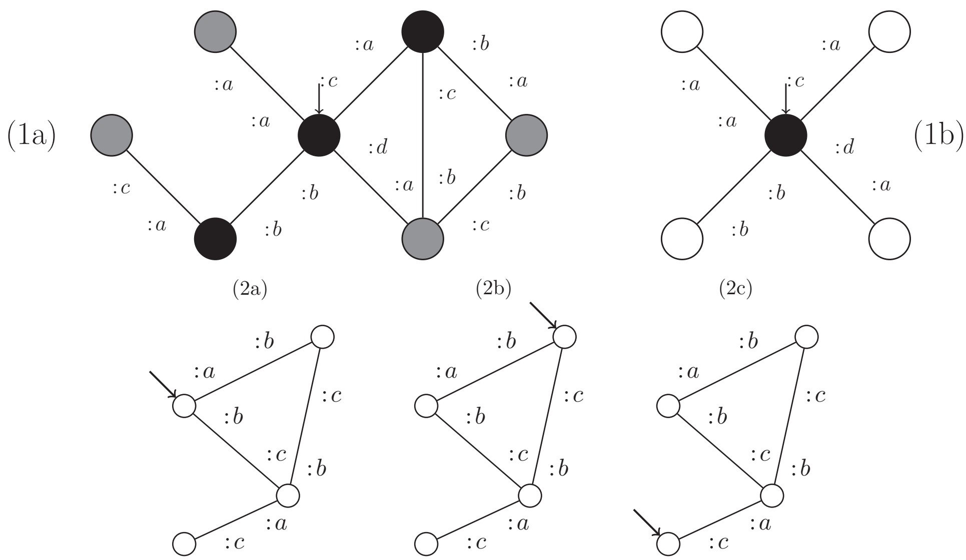


Fig. 4. *Basic operations over generalized Cayley graphs.* (1) From X to X^0: taking the *subdisk of radius* 0. In general, the neighbours of radius r are just those vertices which can be reached in r steps starting from the origin, whereas the disk of radius r, written X^r, is the subgraph induced by 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.

- a finite set π, whose elements are called ;
- a set $E(G)$ of non-intersecting two element subsets of $V(G):\pi$, whose elements are called edges. In other words, an edge e is of the form $\{u:a, v:b\}$, and $\forall e,e' \in E(G), e \cap e' \neq \emptyset \Rightarrow e = e'$.

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 2.2 (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 the above definition, 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 neighbouring vertices, not the states of the neighbouring vertices. A concrete example of this is given in Section 5 and Figure 4.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

We now want to single out a vertex. The following definition is illustrated by Figure 3b.

Definition 2.3 (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. The next definition of isomorphism formalizes the notion of vertex renaming in a graph.

Definition 2.4 (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 the following definition, we use the notion of isomorphism to get rid of all names in the graph.

Definition 2.5 (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 constitute the set of *configurations* of the generalized CA that we consider in this paper.

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. We now formalize these ideas.

Definition 2.6 (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 travel in 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

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

$v_0, \ldots, v_n \in V(G)$ 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 $v_0 = p$ 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 the path α read backwards. The detailed algebraic structure of the set of paths $L(X)$ of a generalized Cayley graph X is described in Section 7.

Two paths are equivalent if they lead to same vertex.

Definition 2.7 (Equivalence of paths). Given a generalized Cayley graph X, we define the *equivalence of paths* relation $\equiv_X$ 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 is often useful to undo the modulo, i.e. to obtain a canonical instance $(G(X), \varepsilon)$ of the equivalence class X.

Definition 2.8 (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 $\alpha.ab \in L$ 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. Section 7 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 are no longer distinguished. The latter notation is given the meaning of the former. We shall speak of a 'vertex' α in $V(X)$ (or simply $\alpha \in X$);

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

- it follows that '$\equiv_X$' and '$=$' are no longer distinguished. The latter notation is 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 make sure that a rigorous meaning can always be recovered by placing tildes back.

3. Basic operations

3.1. *Operations on generalized Cayley graphs*

Overview. Intuitively, given a generalized Cayley graph X, X^r denotes the subdisk of radius r around the pointer. The pointer of X could be moved along a path u, leading to $Y = X_u$. It could be moved back where it was before, leading to $X = Y_{\overline{u}}$. We could also move the pointer, and then take the subdisk, which we denote X_u^r. The purpose of this subsection is to provide a thorough formalization of these basic operations on generalized Cayley graphs. Yet, Figure 4 summarizes these basic operations, and is in fact sufficient to follow most of the paper.

Subdisks. For a pointed graph (G, p) non-modulo (see Arrighi and Dowek (2012) 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 the following:

Definition 3.1 (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 $u \subseteq u'$. Thus, we 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 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 $u \subseteq u'$. Figure 5 presents a generalized Cayley graph together with its disk of radius 0.

Definition 3.2 (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 $|u| \leqslant r+1$, if and only if $u \in X^r$.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

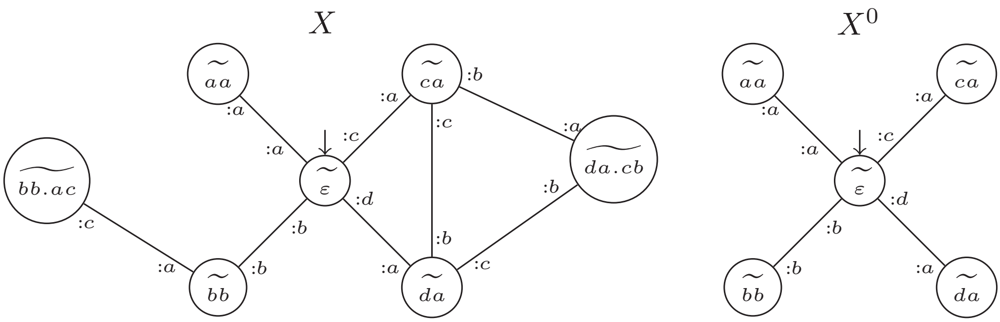


Fig. 5. *A generalized Cayley graph and its disk of radius* 0. Notice that the equivalence classes describing vertices in X^0 are strict subsets of those in X, even though their shortest representative is the same. For instance, the path $ca.cb$ is in $\tilde{da}$ in X but is not a path in X^0, and thus does not belong to $\tilde{da}$ in X^0.

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

Definition 3.3 (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 by u and then a shift by v (i.e. $(X_u)_v$) coincides with a shift by u concatenated with v, and so it may be written $X_{u.v}$.

The composition of a shift, and *then* a restriction, applied on X, is simply written X_u^r. Whilst this is the analogous operation to G_u^r over pointed graphs non-modulo, notice that the shift-by-u completely changes the names of the vertices of X_u^r. As the naming has become relative to u, the disk X_u^r holds no information about its prior location, u.

According to Definition 3.3, $G(X)$ and $G(X_u)$ are isomorphic. Given R such that, $G(X_u) = RG(X)$, the restriction of R^{-1} to $V(G(X_u))$ is uniquely determined; hence, the definition is sound.

It also helps to have a notation for the paths to ε relative to u.

Definition 3.4 (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, 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 need that the vertices of our graphs non-modulo be of a particular form.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

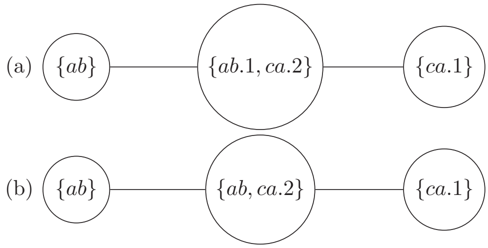


Fig. 6. (a) Is a valid graph as all its vertices names are disjoint subsets. However, (b) is not valid as vertices names $\{ab\}$ and $\{ab, ca.2\}$ intersect.

3.2. *Operations on graphs*

This subsection only becomes useful when tackling the notion of localizability in subsection 5.2, and may be skipped for now. In Section 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, $\{u.1\}$ is 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 6 presents two examples of choices of names for the same graph, one being valid while the other is not.

Definition 3.5 (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.

The next two definitions are standard, see Boehm et al. (1987), Löwe (1993) and Arrighi and Dowek (2012), 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 3.6 (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. $\forall x \in G \, \forall x' \in G' \quad x \cap x' \neq \emptyset \Rightarrow x = x'$,

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

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 $x \cap x' = \emptyset$.

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 are still 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. Figure 7 gives an example of the two operations of shift isomorphism and union.

Definition 3.7 (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_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 (Arrighi and Dowek 2012).

4. Generalized Cayley graphs: 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 CA configurations (Hedlund 1969).

Definition 4.1 (Gromov–Hausdorff–Cantor metrics). Consider the function

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

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

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

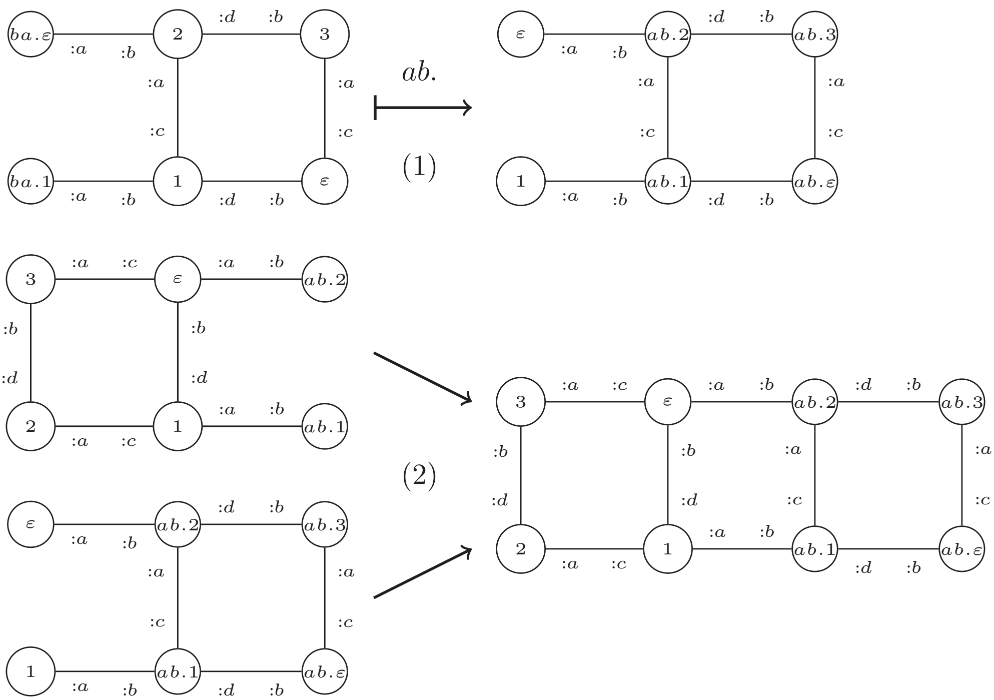


Fig. 7. Operation over graphs. (1) A prefixing of a graph by the word *ab*. The structure of the graph is preserved, only the names of the vertices are changed. (2) A graph union. Here the two graphs on the left-hand side intersect on vertices ε, 1, *ab*.1 and *ab*.2. As the two are consistent (e.g. 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.

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: [Non-negativity, symmetry, identity of indiscernibles] are obvious.
[Equivalence]

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

[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 $k \leqslant l$ so that Y,Z differ only after index k. But then $d(Y,Z) = 1/2^k$ which is $d(X,Z)$.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

[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 4.1 (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 π, 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 (Fedorchuk et al. 1990): Given two $\mathcal{X}$ and $\mathcal{Y}$ be metric spaces 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 CA were first studied in Hedlund (1969), with self-contained elementary proofs available in Kari (2011). For CA over Cayley graphs, a complete reference is Ceccherini-Silberstein and Coornaert (2010). For Causal Graph Dynamics (Arrighi and Dowek 2012), 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 CA. Figure 8 illustrates the notion of continuity in the case of graph dynamics.

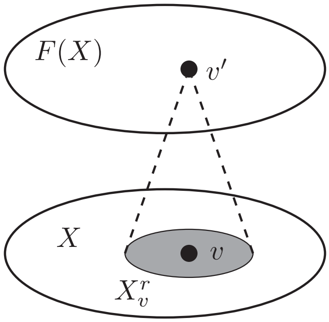


Fig. 8. Continuity of a transformation F. The label and connectivity of vertex v' only depends on the neighbourhood of its antecedent v.

5. Causality and Localizability

5.1. *Causality*

The notion of causality we propose extends the known mathematical definition of CA over grids and Cayley graphs. The extension is a strict one for two reasons: not only the graphs become arbitrary, but they can also vary in time.

The main difficulty we encountered when elaborating an axiomatic definition of causality from $\mathcal{X}_{\Sigma,\Delta,\pi}$ to $\mathcal{X}_{\Sigma,\Delta,\pi}$, was the need to establish a correspondence between the vertices of a generalized Cayley graph X, and those of its image $F(X)$. 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 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 evolution 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 5.1 (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 from $\mathcal{P}(V(X))$ 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 $\{u', v', \ldots\} = R_X(\{u, v, \ldots\}) \subseteq V(F(X))$ end up being marked by a white stone as a consequence of $\{u, v, ... \in X\} \subseteq V(X)$ being marked. Notice that no particular assumption is made on R_X. In particular, R_X can be non-injective, thus effectively allowing vertices to disappear through the dynamics. Now, clearly, the set $\{(X, S) \mid X \in \mathcal{X}_{\Sigma,\Delta,\pi}, S \subset V(X)\}$ is in one-to-one correspondence with $\mathcal{X}_{\Sigma',\Delta,\pi}$ with $\Sigma' = \Sigma \times \{0, 1\}$. Hence, we can define the function F' that maps $(X, S) \cong X' \in \mathcal{X}_{\Sigma',\Delta,\pi}$

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

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 turns out to be very useful.

Definition 5.2 (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 above two conditions are equivalent to just one $F'(X_u) = F'(X)_{R_X(u)}$.

Definition 5.3 (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 m, there exists n such that for all X', $X'^n = X^n$ implies $\operatorname{dom} R^m_{X'} \subseteq V(X'^n)$, $\operatorname{dom} R^m_X \subseteq V(X^n)$ 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.

Finally, our last condition forces the 'expansion' of the graph through F to be bounded. This is done by requiring that any vertex in the image graph is at a bounded distance of the continuation of a vertex of the initial graph.

Definition 5.4 (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 exist $u' \in \operatorname{im} R_X$ and $v' \in F(X)^b_{u'}$ such that $w' = u'.v'$.

The following is our main definition:

Definition 5.5 (Causal dynamics). A dynamics is *causal* if and only if it is shift-invariant, continuous and bounded.

At this point, we introduce several examples that shall appear regularly across the paper to illustrate key notions of our construction. A more complex and physically sound example is developed in the appendix.

Inflating grid. An example of causal dynamics is the inflating grid dynamics illustrated in Figure 9. In the inflating grid dynamics, each vertex gives birth to four distinct vertices, forming a square. All those squares are glued together to form a new graph whose structure respects the initial structure. In other words, the image graph looks like 'a scaled

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

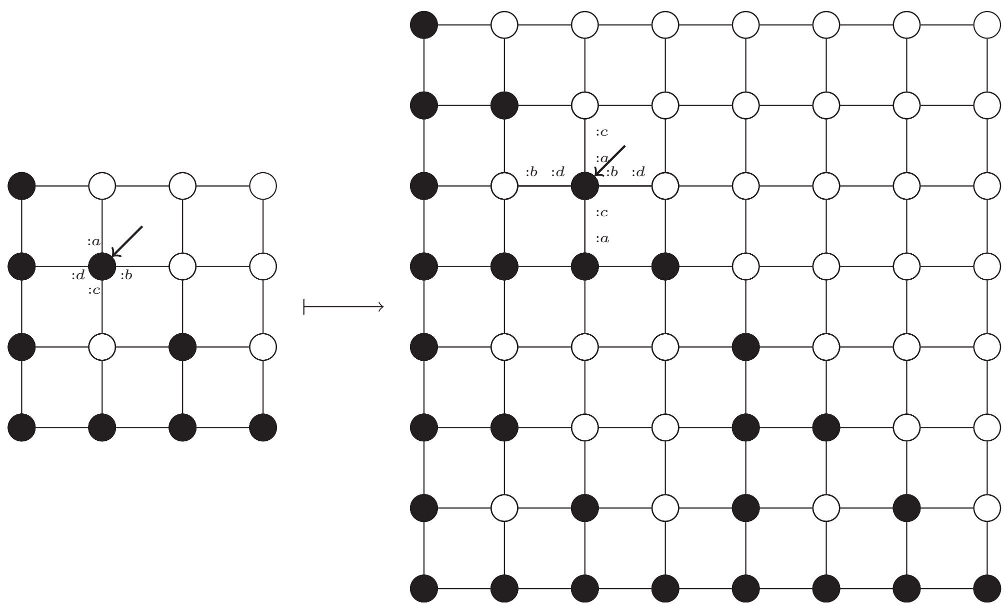


Fig. 9. *The inflating grid dynamics.* Each vertex splits into four vertices. The structure of the grid is preserved. For this precise graph, all edges are connected to ports as stipulated on the pointed vertex (port $:a$ on top, $:b$ on the right, $:c$ on the bottom and $:d$ on the left).

by a factor of 2' version of the initial graph. Graph have maximal degree 4, and the set of ports is $\pi = \{a, b, c, d\}$. Moreover, vertices carry a label in $\Sigma = \{$●, ○$\}$. The rule is chosen such that a black vertex gives birth to a square with three black vertices and one white vertex, whereas a white vertex simply gives birth to four white vertices. For this dynamics, the $R_\bullet$ operator is defined as follows:

$$R_X(u_0.u_1.\dots.u_n) = T(u_0).T(u_1).\dots.T(u_n),$$

where T is the function acting on letters in π^2 as given by the following tables:

$u \in \pi^2$	$T(u)$
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

$u \in \pi^2$	$T(u)$
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

Indeed, for any graph X, R_X maps a path in X into the corresponding path in $F(X)$. Since $F(X)$ 'looks' like X up to a scale factor of 2, it is sufficient to define R_X locally, just

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

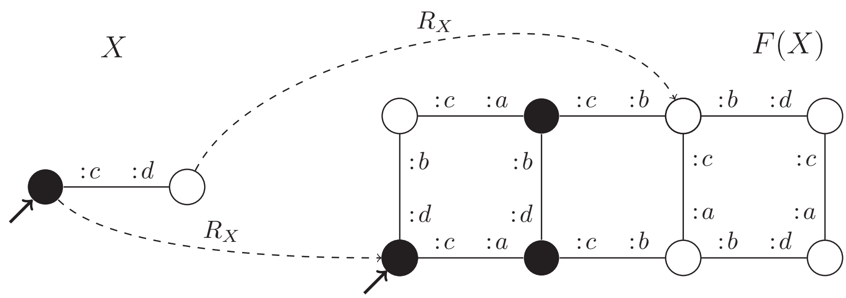


Fig. 10. To each original vertex of X, R_X associates a vertex of $F(X)$, within the corresponding square of four vertices. Here, from the point of view of the vertex on the left, the right vertex is on port c, meaning it sees it as being 'south.' Moreover, it sees a port b on the other end the edge, meaning that the right vertex sees him as being 'west.' By convention, the (newly generated) pointed vertex always lie in the 'north-west'corner of the square. Therefore, the path linking the two pointed vertices in $F(X)$ starts by reaching the 'south' side of the left square (i.e. the side where ports c are connected, which corresponds to an initial path ca), cross the edge between the two squares (i.e. the cb edge), then reach the 'north-west' side of the right square (which corresponds to a path ac). Hence, we have $R_X(cd) = T(cd) = ca.cb.ac$.

by specifying how it maps a path of length 1 (i.e. an edge) of X into a path of $F(X)$. This is what T does.

Since there are four different ports, T must be defined for all combination of two ports forming an edge, hence the 16 cases in the tables above. The apparent complexity of T is due to the fact that among the four vertices of each square of the image graph, only one can be identified with the old vertex of the ancestor graph that gave rise to them. Thus, T applied to an edge must produce a path starting from the old vertex of the square and leading to the old vertex of the neighbouring square. Sometimes this path has to cross both squares, as in the case of $T(cc)$, whereas, in other cases, the two vertices are already next to each other, as in $T(da)$. Figure 10 presents R_X for a given X.

Remark: One way to obtain a simpler dynamics is to restrict the set of configurations where port a always faces port c and port b always faces port d. This ensures that any configurations will, at least locally, look like a grid. As this subset of configurations is compact, and stable by application of the dynamics, all the results of this paper carry through. Under this restriction, the T described boils down to four cases: $T(ac) = ac.ac$, $T(ca) = ca.ca$, $T(bd) = bd.bd$ and $T(db) = db.db$.

The turtle dynamics. The turtle dynamics acts on graphs of degree at most one, without labels: $\Sigma = \emptyset$ and $\pi = \{0\}$. Since our graphs are connected, there exist only two distinct graphs of degree 1. The dynamics simply exchange the two graphs at each time step: The graph composed of a lonely vertex, X_0, becomes an edge connecting two vertices, X_1, and vice-versa. The $R_\bullet$ operator for this dynamics is simple. R_{X_0} is only defined on the empty path (the only path of X_0) and returns the empty path. R_{X_1} can act on the empty path to produce the empty path, and on the path 00 to produce the empty path again. Figure 11 depict the turtle dynamics and its associated $R_\bullet$ operator.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

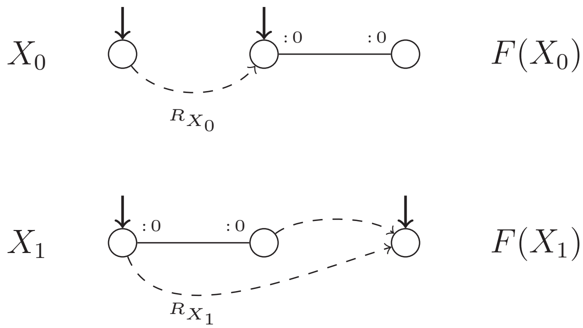


Fig. 11. The turtle dynamics is entirely specified in this figure, since there are only two distinct graphs in its configuration space. Dashed arrows represent the $R_\bullet$ operator. In both cases, R_X maps every path of the graph to the empty path.

Finally, it is possible to show the following result on the growth speed of paths through the application of a map $R_\bullet$ associated to a causal dynamics.

Lemma 5.1 (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 $|R_X(u)| \leqslant (r+1)b$.

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 modulo, 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. □

5.2. *Localizability*

The notion of localizability of a function F captures the exact same idea as the constructive definition of a CA, namely that F arises as a single local rule f applied synchronously and homogeneously across the input graph. Its formalization crucially relies on the conventions, notations and operations of Subsection 3.2.

The general idea is that the local rule f looks at part 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 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 3.2.

Definition 5.6 (Dynamics non-modulo). Consider a function f from $\mathcal{X}^r_{\Sigma,\Delta,\pi}$ to $\mathcal{G}_{\Sigma,\Delta,\pi}$. It is said to be a *dynamics* if and only if for all X the vertices of $f(X)$ are disjoint subsets of $V(X).S$, where S is a finite set of the form $S = \{\varepsilon, 1, \ldots, d\}$ for some integer d and $\varepsilon \in f(X)$.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

Intuitively, the integer $z \in S$ stands for the 'successor number z.' Hence, the vertices designated by $\{1\}, \{2\} \dots$ are successors of the vertex ε, whereas $\{\varepsilon\}$ is its 'continuation.' The vertices designated by $\{ab.1\}, \{ab.2\} \dots$ 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 $V(X)$.

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

Definition 5.7 (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.

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

Definition 5.8 (Local rule). A function f from $\mathcal{X}^r_{\Sigma,\Delta,\pi}$ 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 12. The only subtlety in the above definition 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.

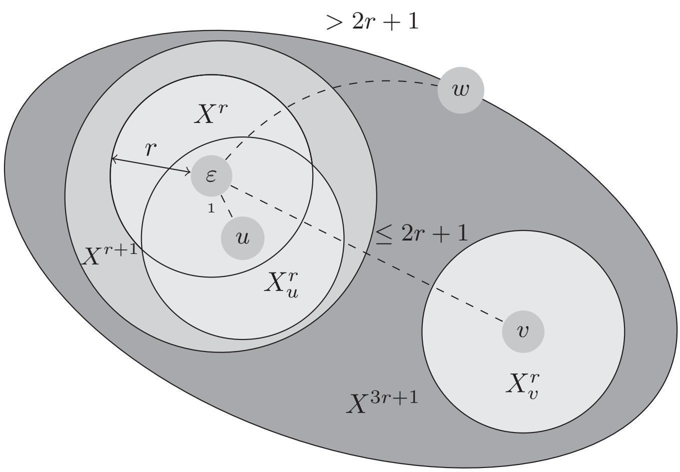


Fig. 12. *The consistency conditions for a local rule.* The drawing represents disks of a generalized Cayley graph X upon which a local rule f of radius r is applied. $f(X^r)$ and $u.f(X^r_u)$ have to be non-trivially consistent since ε and u are at distance 1. $f(X^r)$ and $v.f(X^r_v)$ have to be consistent but their intersection is allowed to be empty. $f(X^r)$ and $w.f(X^r_w)$ are trivially consistent as they are too far to interact in one time step. The disk X^{r+1} is enough to check all the non-trivial consistency conditions, as it comprises first neighbours and their r-disks. The disk X^{3r+1} is enough to check all the consistency conditions, as it comprises all the $2r+1$ neighbours and their r-disks.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

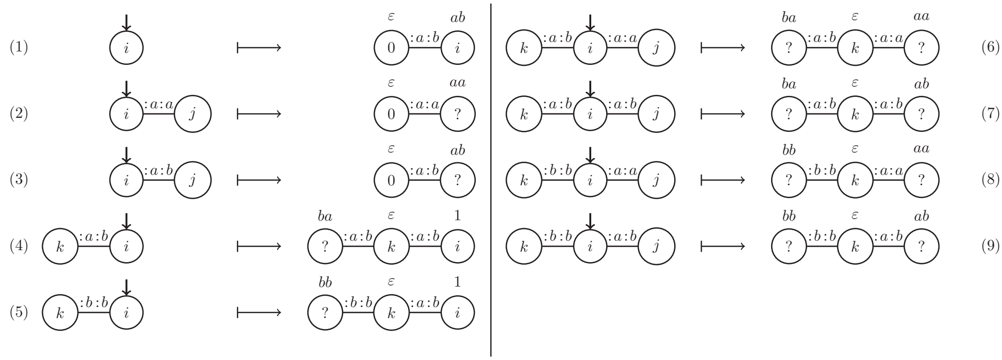


Fig. 13. A way to complete the local rule of the shift cellular automaton into a localizable graph dynamics.

consistent and overlapping, see Figure 12. The point here is to enforce the connectedness of the union of the pieces of graphs via a local, syntactic restriction.

To illustrate the concept of local rule, we now describe local rules inducing a classical CA, the inflating grid and the turtle dynamics introduced in Section 5.1.

Cellular automata. First notice that, because causal graph dynamics are strictly more powerful than CA, it is a tricky task to associate to a given CA a single, canonical causal graph dynamics. Consider, for instance, a one-dimensional CA with internal state in a finite set Σ. The corresponding set of graphs would be the set of graphs with vertex labels in Σ and with ports $\pi = \{a, b\}$. The graphs of this set are (potentially finite) lines. Not all of these lines can be interpreted as CA configurations: They have to be infinite and to have edges of the form *ab* (i.e. no edges of the form *aa* or *bb*). In fact, if one tries to transform a given local rule inducing our CA into a local rule for a corresponding causal graph dynamics, one must choose what to do when the neighbourhood is incomplete (i.e. when the rule acts on a vertex on the edge of a finite line) or ill-formed (i.e. when there is a *aa* or *bb* edge in the vicinity of the vertex). There are two distinct ways of tackling this issue:

— The canonical approach. Consider the subset of graphs composed by the well-formed infinite graphs of our configuration set. This subset has the nice property of being a compact subset. Even better, it is isomorphic, as a metric space, to the set of configurations of the initial CA. We can simply construct a local rule that will replicate the topology of the neighbourhood it sees, and change the labels of the vertices according to the local rule of the CA. This approach is canonical since it requires no additional choice.

— The non-canonical approach simply consists in completing the local rule by choosing what to do when the neighbourhood is ill-formed or incomplete.

Figure 13 depicts a choice of rule completion for the shift automaton. The shift automaton is the one-dimensional cellular automaton that simply shifts the configurations one step in the right direction. Among all the possible neighbourhoods in the causal dynamics version of the shift, only one, up to internal states, is well-formed (the seventh neighbourhood).

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

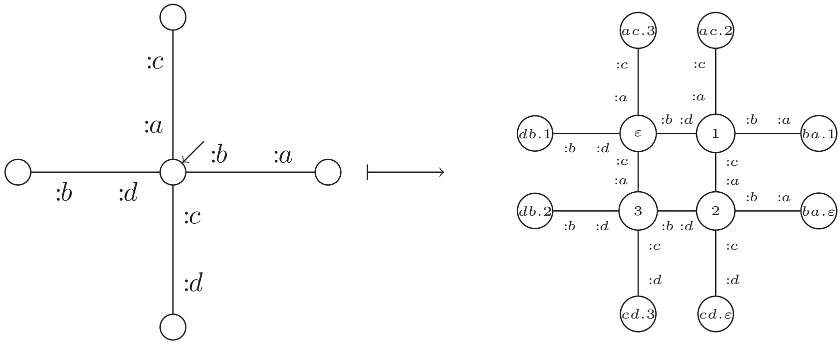


Fig. 14. *Standard case of the inflating grid local rule.* The left-hand side of the rule is a generalized Cayley graph of form X_u^0 (a disk of radius 0). The right-hand side is a graph whose vertex names are subsets of $V(X_u^0).S$. Here they are just singletons, curly brackets are dropped: e.g. we wrote $ac.3$ for $\{ac.3\}$, which should be understood as 'the third successor of my neighbour on edge ac.'

For all the other neighbourhoods, one must decide what to do precisely. Here, we chose the following approach: When the vertex has no right neighbour (i.e. no neighbour on its port b), it creates a new vertex and shift its state inside it (see cases 1, 4 and 5). When the vertex has no left neighbour (i.e. no neighbour on its port a), it assumes that an internal state 0 (white) is coming from the left and thus switch to state 0 while still propagating its internal state to the right (see cases 1, 2 and 3). The canonical completion for the same cellular automaton would simply consists in the case 7, as all the other cases would never appear in the configuration subset we would consider.

Inflating grid. The local rule is of radius zero: it 'sees' the neighbouring vertices and nothing more. In order to be exhaustive, we would need to list all possible neighbourhoods, there are finitely many of those, and, for each of them, specify the corresponding generated subgraph. However, in the standard case, the local rule is applied on a vertex surrounded by four neighbours. It then generates a graph of 12 vertices, each with particular names (see Figure 14). 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 15). 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 16).

Definition 5.9 (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 class modulo isomorphism, of the pointed graph

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

with ε taken as the pointer.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

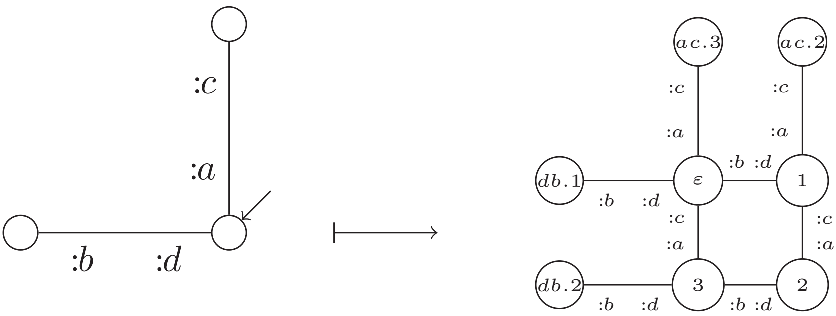


Fig. 15. A particular case of the inflating grid local rule.

5.3. *Equivalence theorem*

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

Theorem 5.1 (Causal is 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.

Sketch of the proof: **[Loc.⇒Caus.]** We consider a localizable function from $\mathcal{X}_{\Sigma,\Delta,\pi}$ to $\mathcal{X}_{\Sigma,\Delta,\pi}$ induced by a local rule f, and prove it must be a continuous, translation invariant and bounded dynamics. Dynamicity and translation invariance in obtained by noticing that each generated subgraph $f(X_u^r)$ has a unique vertex named ε (i.e. whose name, as a set of paths, contains ε). This implies that the subgraph $u.f(X_u^r)$ has a unique vertex named u. Hence, we can build a functional relation between the vertices u in X and the vertices of $F(X)$. This defines an $R_\bullet$ operator which we can show is translation invariant.

Boundedness is easy to check: Since the local rule is finite, the size of the generated subgraphs is bounded, which bounds the growth speed.

Continuity is obtained by considering the worst case scenario where the local rule collapses all the neighbourhoods as much as it is allowed to.

[Loc.⇐Caus.] Here, we need to start from a causal dynamics and build a local rule inducing it. Using continuity, we can extract subgraphs around each vertex reached by the origin (i.e. around each $R_X(u)$ for $u \in X$) that will constitute the image of the local rule. Continuity ensures over this compact space ensures uniform continuity, which ensures that for large enough subgraphs, we will find a functional dependance between neighbourhoods in X and such subgraphs. The technicality of this process lies in the naming choices for the vertices in the image graphs of the constructed local rule.

In the proof of Theorem 5.1, 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

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

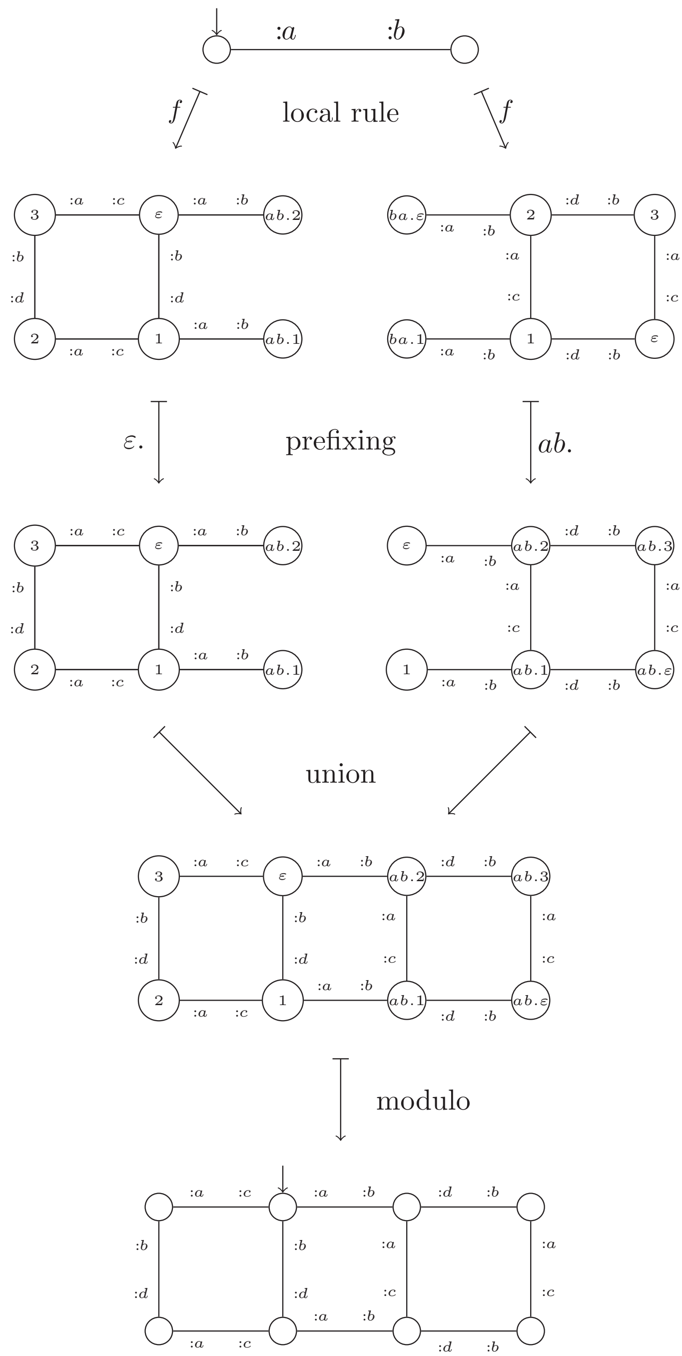


Fig. 16. *Local rule implementation of the inflating grid dynamics.* First, the local rule is applied on the neighbourhood of every vertex of the input graph. The obtained graphs are prefixed (see Definition 3.5) by the vertex they are issued of. Third, a union of graphs is performed to obtain the output graph. Last, the corresponding pointed graph modulo is returned.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

path between u' and w'. The following lemma formalizes this idea as well as some useful, technical although expected properties.

Lemma 5.2 (Local renaming properties). Let $(F, R_\bullet)$ be a causal dynamics. Let b be the maximum of the bounds from Definition 5.4 and Lemma 5.1. 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 $R_X^m = R_{X'}^m$, from Definition 5.5 and Heine's Theorem. We denote $V(\mathcal{X}_\pi^b) = \bigcup_{X\in\mathcal{X}_\pi^b} V(X)$. Let z be an arbitrary injection from $V(\mathcal{X}_\pi^b)\backslash\varepsilon$ 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 $|v'| \leqslant b + 1$. 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 $R_X^m = R_{X^r}^m$, we have that

$$u' = R_X(u) \Leftrightarrow u' = R_X^m(u) \Leftrightarrow u' = R_{X^r}^m(u) \Leftrightarrow u' = R_{X^r}(u),$$

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

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

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 $V(X)$, 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 $w' \in F(X_{u.v})^b$.

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

□

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

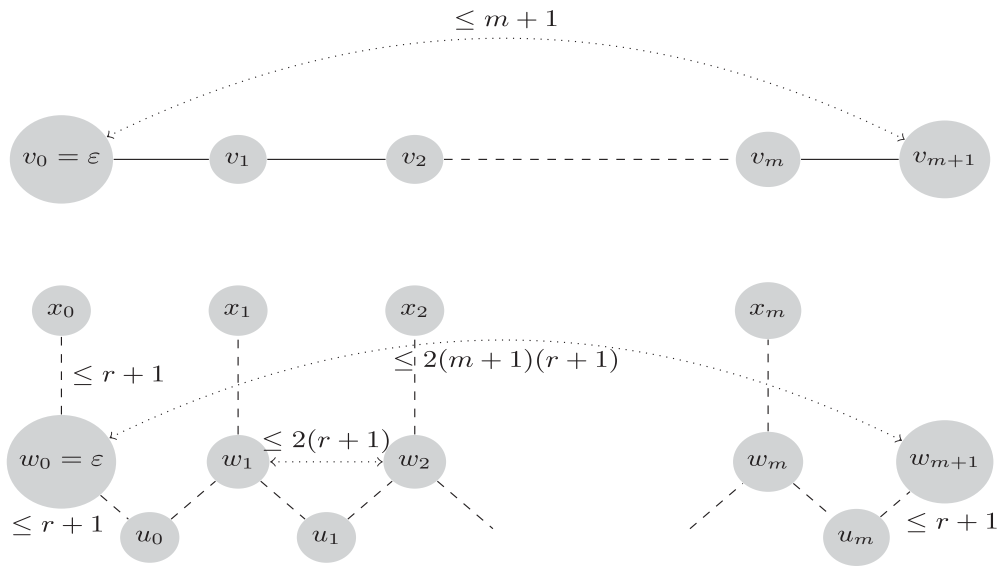


Fig. 17. Proof of continuity.

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 $H(X) = \bigcup u.f(X^r_u)$.

[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 some distance b of u in $H(X)$. Since $G(F(X) = RH(X)$, w' lies at most at a some distance b of $u' = R(u) = R_X(u)$ in $F(X)$.

[Continuity] The following is illustrated in Figure 17. Let $m \in \mathbb{N}$. We must show that there exists n such that $F(X)^m = \tilde{H}(X)^m_\varepsilon$ is determined by X^n.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

Consider a sequence $v_0 = \varepsilon, v_1, \dots v_{m+1}$ of vertices of $H(X)$ such that for all $i \in \{0 \dots m\}$ 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 given Definitions 3.6 and 3.7, 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 $u_0, u_1, \dots u_m$ 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 $w_0 = \varepsilon, w_1, \dots w_{m+1}$ 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 $\bigcup_{i=0\dots m} u_i.X^r_{u_i} \subseteq X^{2(m+1)(r+1)-1}$. 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 $x_0, x_1, \dots, x_m$ 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 $\bigcup_{i=0\dots m} x_i.X^r_{x_i} \subseteq X^{2m(r+1)+2r+1}$. 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, $\mathrm{dom} R^m_X \subseteq X^n$. 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 5.4 and Lemma 5.1, 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 5.5 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 5.2 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 5.2 (*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 5.2 (*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 5.2 for this purpose, i.e. $H(X) = S_X G(F(X))$.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

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 5.2 (*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 5.2 (*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 representant 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 the one given by Definition 5.4, insuring that all the vertices and edges of $F(X^r)$ are covered, along with their labels. □

5.4. *Computability*

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 5.1 (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 5.2 (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 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. □

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

6. Properties

6.1. *Composability*

We 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 that it is stable under composition.

Definition 6.1 (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 CA, but has failed to be obtained for the early definitions of probabilistic CA and quantum CA (see Arrighi et al. (2011), Dürr and Santha (1996), Schumacher and Werner (2004) and Arrighi et al. (2008) for a discussion).

Theorem 6.1 (Composability (Arrighi and Dowek 2012)). Consider two 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 the following:

- $(G \circ F)$ 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 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 n a radius 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 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 $G(F(X_u)) = G(F(X)_{R_X(u)}) = G(F(X))_{S_{F(X)}(R_X(u))}$, $T_X(u.v) = S_{F(X)}(R_X(u.v)) = S_{F(X)}(R_X(u).R_{X_u}(v))) = S_{F(X)}(R_X(u)).S_{F(X)_{R_X(u)}}(R_{X_u}(v)) = T_X(u).S_{F(X_u)}(R_{X_u}(v)) = T_X(u).T_{X_u}(v)$.

[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 5.1 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 $w'' = u''.t''.v''$. □

The above proof was done via the axiomatic characterization of causal dynamics, as this paper enjoys a more straightforward formalization than Arrighi and Dowek (2012). In Arrighi and Dowek (2012), the same result is proven via the constructive approach to causal graph dynamics (localizability), which has the advantage of extra information

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

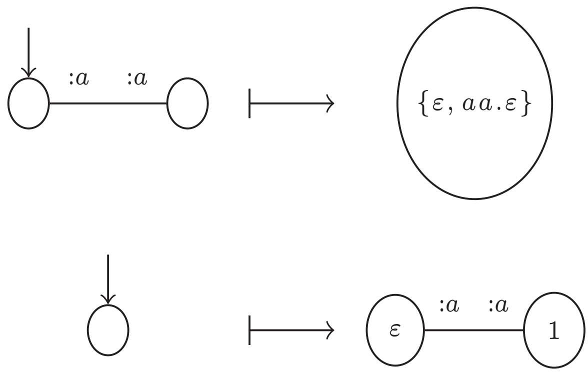


Fig. 18. The local rule inducing the turtle dynamics. There are two distinct neighbourhoods (a vertex with a single neighbour, and a lonely vertex). In the first case, the local rule generates a single vertex with name $\{\varepsilon, aa.\varepsilon\}$. Notice that both vertices will each generate a vertex, and that these two vertices will be identified, since $\{\varepsilon, aa.\varepsilon\} = aa.\{aa.\varepsilon, \varepsilon\}$. In the second case, the local rule generates two vertices linked by an edge. Here, the particular names of those vertices do not matter, so long as they do not intersect. Since the vertex is alone in the graph, only one subgraph is generated, and the subsequent graph union is trivial.

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 $f = de$) from $\mathcal{D}^t$ to $\mathcal{G}_{\Sigma,\Delta,\pi}$ 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.

6.2. *Invertibility, reversibility*

Let us turn our attention to some set-theoretical properties.

Definition 6.2 (Shift-invariant invertible). Consider a shift-invariant dynamics $(F, R_\bullet)$. It is a *shift-invariant invertible* if and only if F is a bijection, and there is an $S_\bullet$ such that $(F^{-1}, S_\bullet)$ is a shift-invariant dynamics.

Notice that there exists some shift-invariant dynamics $(F, R_\bullet)$ such that F is a bijection but there exists no $S_\bullet$ such that $(F^{-1}, S_\bullet)$ is a shift-invariant dynamics (Outline: take $\pi = \{a, b\}$, and map the four-sized directed segment to the seven-sized directed segment pointed on the first four positions and the three-sized directed segment to the seven-sized directed segment but pointed on the last three positions). In this paper, we do not consider them. Notice also that there exists some shift-invariant dynamics $(F, R_\bullet)$ such that $(F^{-1}, S_\bullet)$ is a shift-invariant dynamics, but $S_{F(X)}$ is not the inverse of R_X. The *Turtle example* of Figure 18 illustrates this possibility.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

Again, in this paper, we do not consider them: We restrict ourselves to vertex-preserving invertible dynamics.

Definition 6.3 (Vertex-preserving invertible). A shift-invariant dynamics $(F, R_\bullet)$ is *vertex-preserving invertible* if and only if F is a bijection and for all X we have that R_X is a bijection.

Those are automatically shift-invariant invertible:

Lemma 6.1 (Vertex-preserving invertible is shift-invariant invertible). If $(F, R_\bullet)$ is a vertex-preserving invertible shift-invariant dynamics, 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

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

□

Back to the case of a causal dynamics, the classical question to ask is whether the inverse is also a causal dynamics.

Definition 6.4 (Reversible). A causal dynamics $(F, R_\bullet)$ is *reversible* if and only if it is shift-invariant invertible with inverse $(F^{-1}, S_\bullet)$ a causal dynamics.

The big question is whether the causality of a forward-time causal evolution F, entails that of the backward-time evolution F^{-1}. In other words, is causality stable under inversion? This question was answered positively in the earlier formalism of Arrighi and Dowek (2012), with a more lengthy proof.

Theorem 6.2 (Reversibility). Consider $(F, R_\bullet)$ a causal dynamics. If $(F, R_\bullet)$ is vertex-preserving invertible, then it is reversible, with inverse $(F^{-1}, S_\bullet)$ where $S_Y = (R_{F^{-1}(Y)})^{-1}$.

Proof. For this proof, it is convenient to switch to the $F' : \mathcal{X}_{\Sigma',\Delta,\pi} \to \mathcal{X}_{\Sigma',\Delta,\pi}$ formalism, introduced right after Def. 5.1. Since $F' = (F, R_\bullet)$ is shift-invariant invertible, we have that $F'^{-1} = (F^{-1}, S_\bullet)$ is shift-invariant. Since F' is continuous over the compact space $\mathcal{X}_{\Sigma',\Delta,\pi}$, with $\Sigma' = \Sigma \times \{0, 1\}$, we have that $F'^{-1} = (F^{-1}, S_\bullet)$ is continuous. Since R_X is bijective, so

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

is $S_{F(X)}$, and thus so is S_Y for any Y. Hence, S is surjective and so $(F^{-1}, S_\bullet)$ is bounded with bound 0. □

Discussion: Garden-of-Eden. Another important result in CA theory, and which is related to invertibility questions, is the so-called Garden-of-Eden (a.k.a Moore–Myhill theorem), which states that pre-injectivity (i.e. injectivity over the set of finite configurations) is equivalent to surjectivity (over the set of configurations). This result has been extended to CA over Cayley graphs, provided that the group which induces the Cayley graph has a certain property (it must be amenable), see Ceccherini-Silberstein et al. (2004). Extending this result to a wider class of graphs is impossible for the surjective implies pre-injective part, and is the subject of ongoing research for the pre-injective implies surjective part, see for instance, Gromov (1999).

In the setting of this paper, there are at least two good reasons for the Garden-of-Eden theorem *not* to hold. The first reason is that here, CA have been extended to generalized Cayley graphs, encompassing not just amenable Cayley graphs, but also the non-amenable ones, and may others: In fact, all arbitrary finite degree graphs. The second reason is that here, CA have been extended to time-varying graphs, for which pre-injectivity becomes a much weaker constraint (counting arguments fail as injectivity can be maintained by generating extra vertices, instead of saturating the space of internal states). For instance, a causal dynamics which just adds a vertex to every free port, is injective but not surjective. It could be interesting, however, to look for non-trivial subclasses of causal dynamics for which the Garden-of-Eden still holds.

6.3. *Subclasses of causal dynamics*

Finally, we mention two natural subclasses of graph dynamics. The first is that where only the topology of the graph is evolving, i.e. there is no internal state on vertices nor edges.

Definition 6.5 (No-state a.k.a graph-only dynamics). A dynamics $(F, R_\bullet)$ is a *graph-only a.k.a no-state dynamics* if and only if it is defined over $\mathcal{X}_\pi$, i.e. the graphs carry no internal state.

All of our results apply unchanged in this graph-only setting, as there was nowhere a particular need for an internal state. Moreover, it seems clear that causal graph-only dynamics can simulate general causal dynamics elegantly (see Lemma 1 of Martiel and Martin (2015) for a detailled proof). But it is not so clear whether this still holds in the reversible case, for instance.

The second, dual class is that where only the internal states are evolving, i.e. the dynamics does not change the graph. This is of course is a widely studied case (Boehm et al. 1987; Ceccherini-Silberstein et al. 2004; Derbel et al. 2008; Ehrig and Lowe 1993; Gromov 1999; Gruner 2010; Kreowski and Kuske 2007; Löwe 1993; Papazian and Remila 2002; Taentzer 1996, 1997; Tomita et al. 2002, 2009, 2005).

Definition 6.6 (Graph-preserving a.k.a state-only dynamics). A dynamics $(F, R_\bullet)$ is a *graph-preserving a.k.a state-only dynamics* if and only if for all X, the graphs X and $F(X)$ have the same structure, i.e. they are the same up to labellings σ, δ.

Again all of our results apply unchanged in this state-only setting, as there was nowhere a particular need for changing the topology, although changing the topology is one of the main contributions of this paper. Still, it could be said that the paper does port the Curtis–Hedlund–Lyndon theorem to CA over arbitrary graphs, and not just Cayley graphs, which had not been done. Some results could of course be made tighter for state-only causal dynamics, such as that of the radius of a composition.

This graph-preserving class was defined by demanding that a certain property be preserved by the graph dynamics. Thus, it falls into the broad class of subspace-preserving dynamics.

Definition 6.7 (Subspace-preserving dynamics). Consider $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ a partition of $\mathcal{X}_{\Sigma',\Delta,\pi}$. A dynamics $(F, R_\bullet)$ is a *subspace-preserving dynamics* with respect to the partition if and only if for all i, we have that $X \in \mathcal{X}_i$ implies that $F(X) \in \mathcal{X}_i$.

On the other hand, the no-state class was defined by restricting the definition space of the causal dynamics. Thus, it falls into the broad class of subspace-restricted dynamics.

Definition 6.8 (Subspace-restricted dynamics). Consider $\mathcal{Y} \subseteq \mathcal{X}_{\Sigma',\Delta,\pi}$. A dynamics $(F, R_\bullet)$ is a *subspace-restricted dynamics* with respect to $\mathcal{Y}$ if and only if its definition is restricted to $\mathcal{Y}$.

In both of these broad classes, it seems cautious to demand that the subspaces be themselves compact spaces, as was the case with the no-state and graph-preserving classes. Indeed, most of the structural results of this paper stringly rely on the compactness of the configuration space (especially Theorem 5.1). Dynamics over non-compact subspaces could probably defined with an ad-hoc construction. Finally, let us mention that in our study of reversibility, we required that our causal dynamics $(F, R_\bullet)$ be vertex-preserving, i.e. that R_X be a bijection between $V(X)$ and $V(F(X))$. This differs from graph-preservation: the connectivity may vary. It is not clear whether this vertex-preserving class could have been defined through a subspace-preservation construction.

7. Adjacency structures

The sole purpose of this last section is to deepen the notion of generalized Cayley graphs. we provide an alternative, more algebraic definition of them, as a language endowed with a theory of equivalence. We then prove the equivalence between this definition and the definition given in Section 2.

7.1. *Paths structures*

Definition 7.1 (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 $S(\mathcal{X}_\pi)$.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

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

Proposition 7.1 (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 $X = Y$. □

7.2. *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 7.2 (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 to make 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 'If one of two equivalent paths admits an extension, then so does the other path, and both extensions will be equivalent'; (*iii*) means that 'if a step takes you from A to B, the inverse step takes you from B to A.'

Definition 7.3 (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

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

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

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 language L and equivalence relation $\equiv$. The *set of all adjacency structures* is written $\mathcal{S}_\pi$. From now on, S represents an element of $\mathcal{S}_\pi$.

The added adjacency structure condition aims to make 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 7.4 (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 7.1 (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 $b = c$. □

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 7.2 (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 $(L \cup L')^2$.

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

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

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 7.3, there exists a word in L describing the path ω', hence $w' \in L$. Similarly, we prove the inclusion $\equiv_{L'} \subseteq \equiv_L$.

7.3. *Graphs as languages*

Generalized Cayley graphs. Summarizing, $S(.)$ is bijective from Proposition 7.1 and $S \circ X = Id$ from Proposition 7.2, thus X is bijective, i.e. the following theorem comes out as a corollary:

Theorem 7.1 (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 7.3 (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 (Godsil et al. 2001), see Figure 19.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press

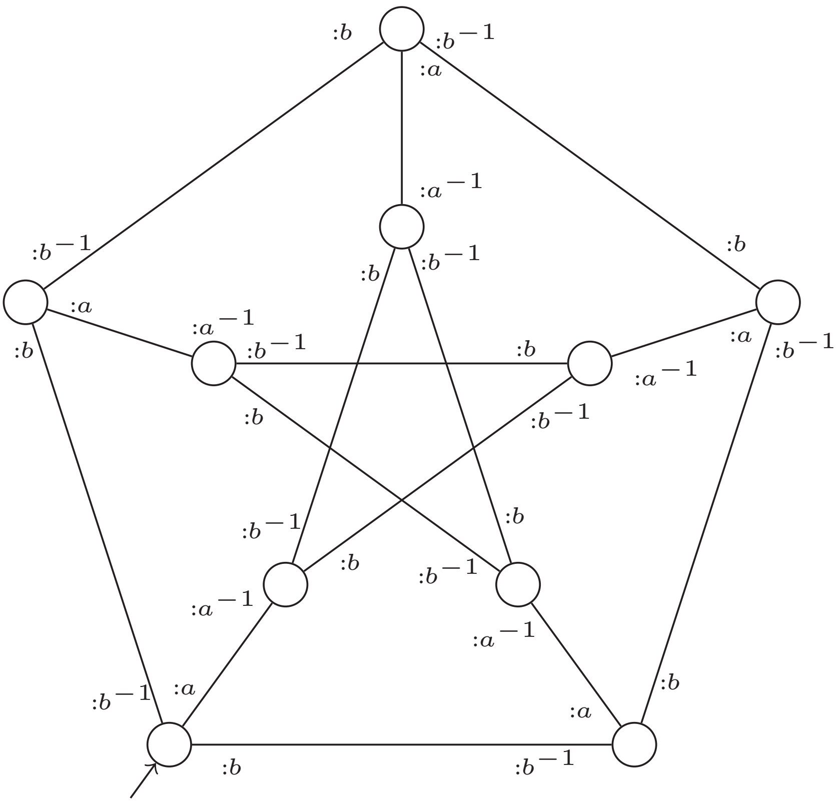


Fig. 19. The Petersen graph as a generalized Cayley graph structure.

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 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 3. We can define a distance between theses graphs, which makes $\mathcal{X}_{\Sigma,\Delta,\pi}$ a compact metric space, as done in Section 4.

8. Conclusion

Summary. First, we have shown that a notion of graphs with ports modulo isomorphism (Definitions 2.1–2.5) provides a generalization of Cayley graphs, in the following sense: Each vertex can be named relatively to the origin; each graph represents a language and its equivalence relation (Definitions 7.2–7.3, Theorem 7.1); and they are equipped with a well-defined notion of translation (Definition 3.3). Second, we have shown that the set of these graphs forms a compact metric space (Definition 4.1 and Lemma 4.1), entailing that continuous functions over this set are also uniformly continuous (Heine's theorem). Third, this allowed us to characterize CA over those generalized Cayley graphs as the set of shift-invariant, continuous, bounded dynamics (Definitions 5.1–5.5). This physically motivated mathematical definition would have remained excessively abstract without our main result, showing that such causal dynamics are necessarily localizable, i.e. that they can be expressed as the synchronous, homogeneous application of a local rule (Definitions 5.6–5.9, Theorem 5.1). Fourth, we showed that the property of being a local rule is decidable and hence that causal dynamics are computable (Propositions 5.1–5.2). Finally, we showed that the composition of two causal dynamics is itself a causal dynamics (Theorem 6.1), and that the shift-invariant inverse of a causal dynamics is again a causal dynamics (Theorem 6.2).

Further works. The mathematical relation between the causal dynamics of Arrighi and Dowek (2012) and ours remains to be clarified – for instance, decidability remains to be proven for the causal graph dynamics of Arrighi and Dowek (2012). Still, they are important features of models of computation. The fact that they are relatively straightforward to prove in this paper is a good indicator that the formalism presented is appropriate.

Our short terms plan, however, include interpreting CA over generalized Cayley graphs as a dynamics over simplicial complexes as was started in Arrighi et al. (2014); deepening the study of the reversible case; formalizing the stochastic case. Moreover, one of the authors has been studying the intrinsic simulation and intrinsic universality of causal graph dynamics in Martiel and Martin (2013), an approach which can still be taken further.

This work has been funded by the ANR-10-JCJC-0208 CausaQ grant, by the John Templeton Foundation grant ID 15619, and by the Stic-Amsud grant FoQCoSS. It also benefited from discussions with Christophe Crespelle, Gilles Dowek, Emmanuel Jeandel, Viv Kendon, Jean Mairesse, Bruno Martin, David Meyer, Simon Perdrix and Éric Thierry. We thank the anonymous referees, who pushed for a better paper.

References

Arrighi, P. and Dowek, G. (2012). Causal graph dynamics. In: *Proceedings of ICALP 2012*, Warwick, Lecture Notes in Computer Science, vol. 7392, 54–66.

Arrighi, P., Fargetton, R., Nesme, V. and Thierry, E. (2011). Applying causality principles to the axiomatization of probabilistic cellular automata. In: *Proceedings of CiE 2011,* Sofia, Lecture Notes in Computer Science, vol. 6735, 1–10.

Arrighi, P., Martiel, S. and Wang, Z. (2014). Causal dynamics over discrete surfaces. In: Ayala-Rincón, M., Bonelli, E. and Mackie, I. (eds.) *Proceedings 9th International Workshop on Developments in Computational Models*, Buenos Aires, Argentina, 26 August 2013, Electronic Proceedings in Theoretical Computer Science, vol. 144, Open Publishing Association, 30–40.

Arrighi, P., Nesme, V. and Werner, R.F. (2008). Quantum cellular automata over finite, unbounded configurations. In: *Proceedings of LATA,* Lecture Notes in Computer Science, vol. 5196, Springer, 64–75.

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

Ceccherini-Silberstein, T. and Coornaert, M. (2010). *Cellular Automata and Groups*, Springer Verlag.

Ceccherini-Silberstein, T., Fiorenzi, F. and Scarabotti, F. (2004). 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*, Berlin: de Gruyter, 73–108.

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

Derbel, B., Mosbah, M. and Gruner, S. (2008). Mobile agents implementing local computations in graphs. In: Ehrig, H., Heckel, R., Rozenberg, G. and Taentzer, G. (eds.) *Graph Transformations: 4th International Conference, ICGT 2008*, Leicester, United Kingdom, September 7–13, 2008. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 99–114.

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

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

Fedorchuk, V. V., Arkhangelskiui, A. V. and Pontriagin, L.S. (1990). *General Topology I*, vol. 1, Springer.

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

Godsil, C. D., Royle, G. and Godsil, C. D. (2001). *Algebraic Graph Theory*, vol. 8, Springer-Verlag, New York.

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

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

Hasslacher, B. and Meyer, D. A. (June 1998). 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.*

Hedlund, G. A. (1969). Endomorphisms and automorphisms of the shift dynamical system. *Math. Systems Theory* **3** 320–375.

Kari, K. (2011). Cellular Automata, Lecture notes. Available at: http:users.utu.fijkarica.

Klales, A., Cianci, D., Needell, Z., Meyer, D. A. and Love, P. J. (Oct. 2010). Lattice gas simulations of dynamical geometry in two dimensions. *Physical Review E* **82** (4) 046705.

Kreowski, H. J. and Kuske, S. (2007). Autonomous units and their semantics - The parallel case. In: Fiadeiro, J. L. and Schobbens, P.-Y. (eds.) *Recent Trends in Algebraic Development Techniques: 18th International Workshop, WADT 2006*, La Roche en Ardenne, Belgium, June 1–3, 2006, Revised Selected Papers, Springer Berlin Heidelberg, 56–73.

Kreowski, H. J. and Kuske, S. (2011). Graph multiset transformation: A new framework for massively parallel computation inspired by dna computing. *Natural Computing* **10** (2) 961–986.

Kurth, W., Kniemeyer, O. and Buck-Sorlin, G. (2005). Relational growth grammars – A graph rewriting approach to dynamical systems with a dynamical structure. In: Banâtre, J.-P., Fradet, P., Giavitto, J.-L. and Michel, O. (eds.) *Unconventional Programming Paradigms: International*

Workshop UPP 2004, Le Mont Saint Michel, France, September 15–17, 2004, Revised Selected and Invited Papers, Springer Berlin Heidelberg, 56–72.

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

Martiel, S. and Martin, B. (2013). Intrinsic universality of causal graph dynamics. In: Neary, T. and Cook, M. (eds.) *Proceedings Machines, Computations and Universality 2013,* Zürich, Switzerland, 9/09/2013–11/09/2013, Electronic Proceedings in Theoretical Computer Science, vol. 128, Open Publishing Association, 137–149.

Martiel, S. and Martin, B. (2015). An intrinsically universal family of causal graph dynamics. In: Durand-Lose, J. and Nagy, B. (eds.) *Machines, Computations, and Universality: 7th International Conference, MCU 2015*, Famagusta, North Cyprus, September 9–11, 2015. Proceedings, Springer International Publishing, Cham, 129–148.

Métivier, Y. and Sopena, E. (1997). Graph relabelling systems: A general overview. *Computers & Artificial Intelligence* **16** 167–185.

Papazian, C. and Remila, E. (2002). Hyperbolic recognition by graph automata. In: *Proceedings of the Automata, Languages and Programming: 29th International Colloquium, ICALP 2002*, Málaga, Spain, vol. 2380, Springer Verlag, 330.

Regge, T. (1961). General relativity without coordinates. *Il Nuovo Cimento (1955–1965)* **19** (3) 558–571.

Róka, Z. (1999). Simulations between cellular automata on Cayley graphs. *Theoretical Computer Science* **225** (1–2) 81–111.

Ryszka, I., Paszyska, A., Grabska, E., Sieniek, M. and Paszyski, M. (2015a). Graph transformation systems for modeling three dimensional finite element method: Part i. *Fundamenta Informaticae* **140** (2) 129–172.

Ryszka, I., Paszyska, A., Grabska, E., Sieniek, M. and Paszyski, M. (2015b). Graph transformation systems for modeling three dimensional finite element method: Part ii. *Fundamenta Informaticae* **140** (2) 173–203.

Schumacher, B. and Werner, R. (2004). Reversible quantum cellular automata. ArXiv pre-print quant-ph/0405174.

Sorkin, R. (1975). Time-evolution problem in Regge calculus. *Physical Review D*. **12** (2) 385–396.

Taentzer, G. (1996). *Parallel and Distributed Graph Transformation: Formal Description and Application to Communication-Based Systems*. PhD thesis, Technische Universitat Berlin.

Taentzer, G. (1997). Parallel high-level replacement systems. *Theoretical Computer Science* **186** (1–2) 43–81.

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

Tomita, K., Kurokawa, H. and Murata, S. (2009). Graph-rewriting automata as a natural extension of cellular automata. In: Gross, T. and Sayama, H. (eds.) *Adaptive Networks*, Understanding Complex Systems, vol. 51, Springer, Berlin/Heidelberg, 291–309.

Tomita, K., Murata, S., Kamimura, A. and Kurokawa, H. (2005). Self-description for construction and execution in graph rewriting automata. In: Capcarrère, M. S., Freitas, A. A., Bentley, P. J., Johnson, C. G. and Timmis, J. (eds.) *Advances in Artificial Life: 8th European Conference, ECAL 2005*, Canterbury, UK, September 5–9, 2005, Springer Berlin Heidelberg. Proceedings, 705–715.

Von Mammen, S., Phillips, D., Davison, T. and Jacob, C. (2010). A graph-based developmental swarm representation and algorithm. In: Dorigo, M., Birattari, M., Di Caro, G. A., Doursat, R., Engelbrecht, A. P., Floreano, D., Gambardella, L. M., Gross, R., Sahin, E., Stützle, Th. and Sayama, H. (eds.) *Swarm Intelligence: 7th International Conference, ANTS 2010*, Brussels, Belgium, September 8–10, 2010, Springer Berlin Heidelberg. Proceedings, 1–12.

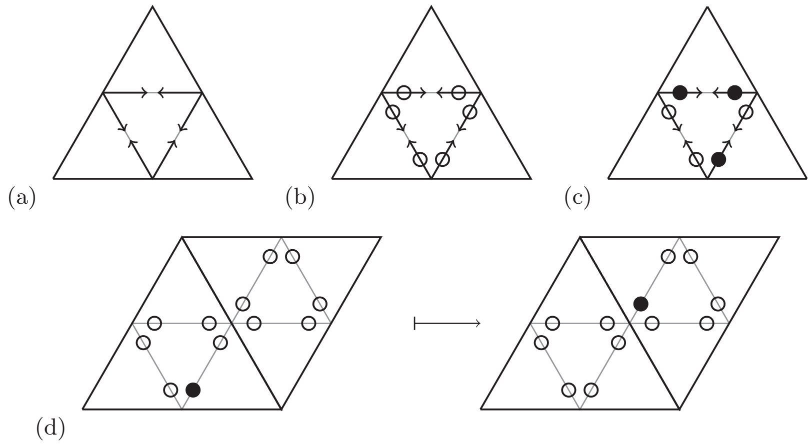


Fig. A1. (a) Each triangle can contain up to six particles, each corresponding to distinct velocity. The presence of a particle of a given velocity is depicted by a filled circle, its absence by an empty circle. (c) Here the triangle hosts three particles moving from left to right, right to left and bottom to right. Arrows are dropped for visibility. (d) Simple particle propagation. In the case where there is no collision pattern, or if they are side-by-side, not much happens: Each particle moves freely to the next triangle in its trajectory, and preserves its velocity.

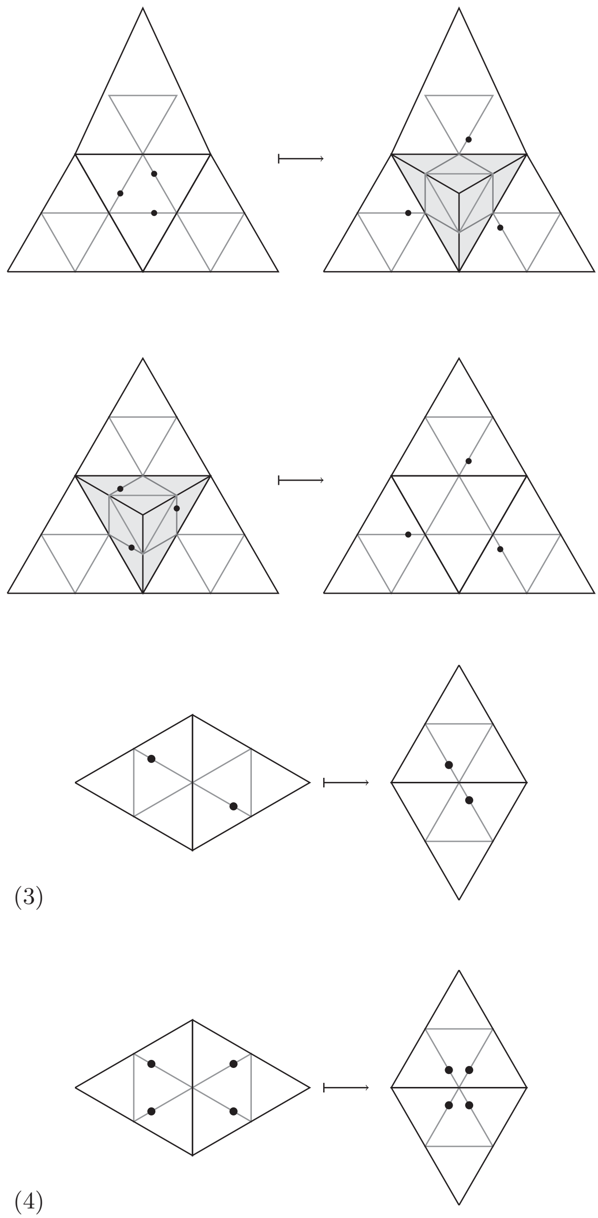
Appendix. Lattice Gas Over Dynamical Geometry

This appendix is dedicated to the introduction of a more detailed example. This dynamics is based on a particular instance of Lattice Gas Automaton where particles can create or delete cells depending on the way they collide. The asymptotical behaviour of this dynamics was thoroughly studied by Meyer et al. both in the one-dimensional case (Hasslacher and Meyer 1998) and in the two-dimensional case (Klales et al. 2010). We present here a two-dimensional implementation of their dynamics using a causal graph dynamics.

Informal presentation of the dynamics. In the original example, the 'lattice' is really a two-dimensional simplicial complex, i.e. triangles that are glued together along their faces to form a discrete surface. Each triangle potentially hosts 0 to 6 particles, each with a fixed, distinct velocity (see Figure A1). At each time step, particles just propagate according to their velocities. Except in four distinguished cases of collision patterns, depicted in Figure A2, and provided that these patterns do not happen next to each other.

Local rule. In order to encode these configurations inside a graph, we simply encode each triangle into a vertex of degree 3: $\pi = \{0, 1, 2\}$. Each glueing is then encoded into an edge inside the graph. Internal labels are used to encode the presence of particles: $\Sigma = \{0, 1\}^6$. In order to ensure that collision patterns are not side-by-side, we need to set the radius of the local rule to at least 2. Figure A3 does not show radius 2, but only the radius 0, to show the non-trivial part of the local rule's action, assuming that the graph transformation is triggered. In other just in order to detect side-by-side collision patterns and thus avoid inconsistent geometrical transformations to occur next to each other.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press



(1)

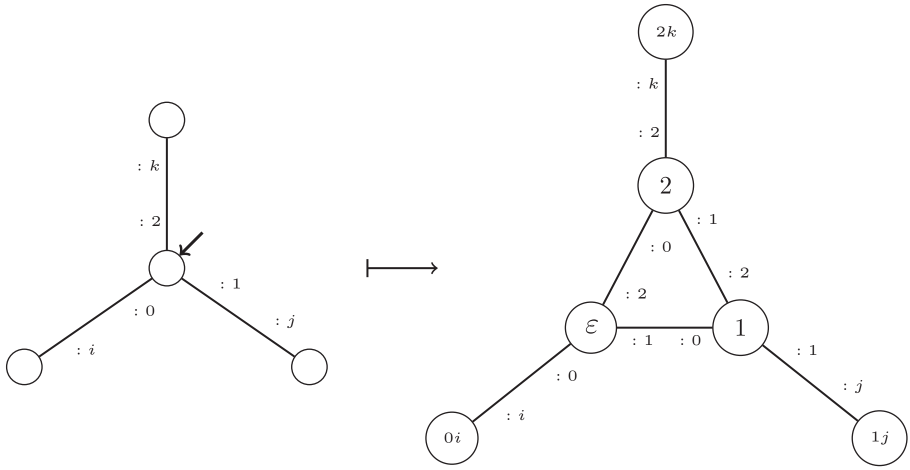
(2)

(3)

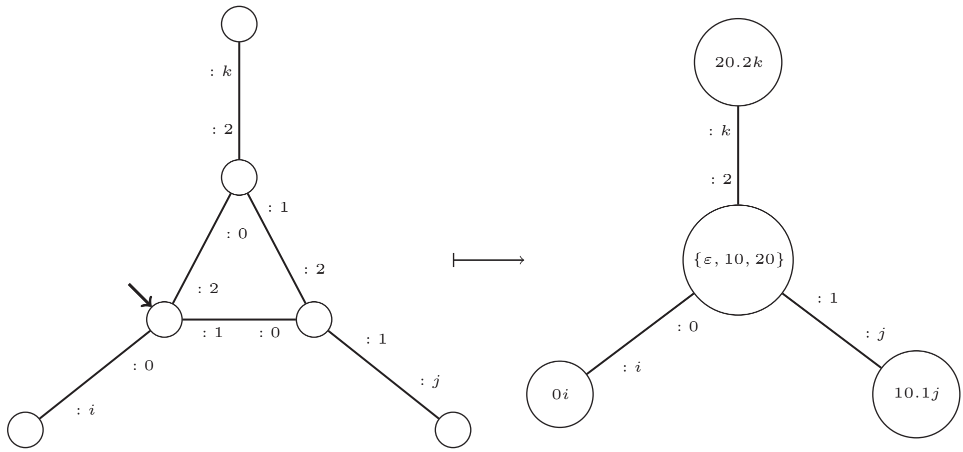
(4)

Fig. A2. Certain collision patterns, when they are isolated, trigger special behaviours. (1) When three particles collide inside a triangle, the triangle is split into three new triangles. (2) Conversely, when three particles cross the faces of three adjacent triangles, the three triangles merge to form a new triangle. (3), (4) When particles meet at a face between two triangles, the face is rotated to create two triangles as depicted.

https://doi.org/10.1017/S0960129517000044 Published online by Cambridge University Press



(a)



(b)

Fig. A3. In all these examples, we assume that the vertices carry labels that indicate the presence of particles, and that these form the one collision pattern that triggers this corresponding geometric transformation. We also assume that this collision pattern is found isolated, i.e. not side-by-side with another. (a) In this case, the local rule splits the centre vertex into three vertices, naming them ε, 1 and 2. The other vertices are untouched and are simply attached to the new vertices. (b) Here, the reverse transformation is performed. The three middle vertices are identified in a single vertex names $\{\varepsilon, 10, 20\}$. Keep in mind that three similar graphs will be generated by the three middle vertices. These graphs will be identical and thus identified during the graph union. (c) In this case, the two centre vertices simply redistribute their edges, no vertex is created.