



**HAL**  
open science

# Causal Graph Dynamics and Kan Extensions

Luidnel Maignan, Antoine Spicher

► **To cite this version:**

Luidnel Maignan, Antoine Spicher. Causal Graph Dynamics and Kan Extensions. 2024. hal-04511147v1

**HAL Id: hal-04511147**

**<https://hal.science/hal-04511147v1>**

Preprint submitted on 19 Mar 2024 (v1), last revised 23 Dec 2024 (v2)

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

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

# Causal Graph Dynamics and Kan Extensions

Luidnel Maignan and Antoine Spicher

Univ Paris Est Creteil, LACL, 94000, Creteil, France

**Abstract.** On the one side, the formalism of Global Transformations comes with the claim of capturing any transformation of space that is local, synchronous and deterministic. The claim has been proven for different classes of models such as mesh refinements from computer graphics, Lindenmayer systems from morphogenesis modeling and cellular automata from biological, physical and parallel computation modeling. The Global Transformation formalism achieves this by using category theory for its genericity, and more precisely the notion of Kan extension to determine the global behaviors based on the local ones. On the other side, Causal Graph Dynamics describe the transformation of port graphs in a synchronous and deterministic way and has not yet being tackled. In this paper, we show the precise sense in which the claim of Global Transformations holds for them as well. This is done by showing different ways in which they can be expressed as Kan extensions, each of them highlighting different features of Causal Graph Dynamics. Along the way, this work uncovers the interesting class of Monotonic Causal Graph Dynamics and their universality among General Causal Graph Dynamics.

## 1 Introduction

*Initial Motivation.* This work started as an effort to understand the framework of Causal Graph Dynamics (CGD) from the point of view of Global Transformations (GT), both frameworks expanding on Cellular Automata (CA) with the similar goal of handling dynamical spaces, but with two different answers. Indeed, we have on the one hand CGD that have been introduced in 2012 in [1] as a way to describe synchronous and local evolutions of labeled port graphs whose structures also evolve. Since then, the framework has evolved to incorporate many considerations such as stochasticity, reversibility [3] and quantumness [2]. On the other hand, we have GT that have been proposed in 2015 in [5] as a way to describe synchronous local evolution of any spatial structure whose structure also evolves. This genericity over arbitrary kind of space is obtained using the language of category theory. It should therefore be the case that CGD are a special case of GT in which the spatial structure happens to be labeled port graphs. So the initial motivation is to make this relationship precise.

*Initial Plan.* Initially, we expected this to be a very straightforward work. First, because of the technical features of CGD (recalled in the next section), it is appropriate to study them in the same way that CA had been studied in [4].

By this, we mean that although GT uses the language of category theory, the categorical considerations all simplify into considerations from order theory in the case of plain CGD. Secondly, CGD seemed to come directly with the required order theoretic considerations needed to make every thing trivial. It indeed has a notion of sub-graph used implicitly throughout. The initial plan was to write these “trivialities” explicitly, to make sure that every thing is indeed trivial, and then proceed to the next step: quotienting absolute positions out to only keep relative ones, thus actually using categorical features and not only order theoretic ones, as done in [4] for CA.

Section 2.2 recalls how CGD are defined and work. Section 2.3 recalls how Kan extensions and GT are defined and simplifies in the particular case of order theory. Section 3.1 makes clear the strong relation between the two concepts and how this relation leads to the initial ambition.

*Actual Plan.* It turns out that the initial ambition falls short, but the precise way in which it does reveals something interesting about CGD. The expected relationship is in fact partial and only allows to accommodate Kan extensions with CGD which happen to be monotonic. This led us to change our initial plan to investigate the role played by monotonic CGD within the framework. Doing so, we uncover the universality of monotonic CGD among general CGD, thus implying the initially wanted result: all CGD are GT.

Sections 3.2 and 3.3 make clear the relationship between GT and monotonic CGD, while Section 4 shows the universality of monotonic CGD by proposing an encoding transforming any CGD into a monotonic one. Finally, Section 5 discusses this interesting property of CGD.

## 2 Preliminaries

### 2.1 Notations

The definitions in this paper mostly use common notations from set theory. The set operations symbols, especially set inclusion  $\subseteq$  and union  $\cup$ , are heavily overloaded, but the context always allows to recover the right semantics. Two first overloads concern the inclusion and the union of partial functions, which are to be understood as the inclusion and union of the graphs of the functions respectively, *i.e.*, their sets of input-output pairs. A partial function  $f$  from a set  $A$  to a set  $B$  is indicated as  $f : A \rightharpoonup B$ , meaning that  $f$  is defined for some elements of  $A$ . The restriction of a function  $f : A \rightharpoonup B$  to a subset  $A' \subseteq A$  is denoted  $f \upharpoonright A' : A' \rightharpoonup B$ .

### 2.2 Causal Graph Dynamics

Our work strongly relies on the objects defined in [1]. We recall here the definitions required to understand the rest of the paper. Particularly, [1] establishes the equivalence of the so-called *causal dynamics* and *localisable functions*. For

the present work, we focus on the latter. Our method is to stick to the original notations in order to make clear our relationship with the previous work. However, some slight differences exist but are locally justified.

We consider an uncountable infinite set  $\mathcal{V}$  of symbols for naming vertices.

**Definition 1 (Labeled Graphs with Ports).** *Let  $\Sigma$  and  $\Delta$  be two sets, and  $\pi$  a finite set. A graph  $G$  with states in  $\Sigma$  and  $\Delta$ , and ports in  $\pi$  is the data of:*

- a countable subset  $V(G) \subset \mathcal{V}$  whose elements are the vertices of  $G$ ,
- a set  $E(G)$  of non-intersecting two-element subsets of  $V(G) \times \pi$ , whose elements are the edges of  $G$ , and are denoted  $\{u : i, v : j\}$ .
- a partial function  $\sigma(G) : V(G) \rightarrow \Sigma$  labeling vertices of  $G$  with states,
- a partial function  $\delta(G) : E(G) \rightarrow \Delta$  labeling edges of  $G$  with states.

The set of graphs with states in  $\Sigma$  and  $\Delta$ , and ports  $\pi$  is written  $\mathcal{G}_{\Sigma, \Delta, \pi}$ .

A pointed graph  $(G, v)$  is a graph  $G \in \mathcal{G}_{\Sigma, \Delta, \pi}$  with a selected vertex  $v \in V(G)$  called pointer.

In this definition, the fact that edges are non-intersecting means that each  $u : i \in V(G) \times \pi$  appears in at most one element of  $E(G)$ . The vertices of the graph are therefore of degree at most  $|\pi|$ , the size of the finite set  $\pi$ . Let us note right now that the particular elements of  $\mathcal{V}$  used to build  $V(G)$  in a graph  $G$  should ultimately be irrelevant, and only the structure and the labels should matter, as made precisely in Def. 6. This is the reason of the definition of the following object.

**Definition 2 (Isomorphism).** *An isomorphism is the data of a bijection  $R : \mathcal{V} \rightarrow \mathcal{V}$ . Its action on vertices is straightforwardly extended to any edge by  $R(\{u : i, v : j\}) := \{R(u) : i, R(v) : j\}$ , to any graph  $G \in \mathcal{G}$  by*

$$\begin{aligned} V(R(G)) &:= \{R(v) \mid v \in V(G)\}, & E(R(G)) &:= \{R(e) \mid e \in E(G)\}, \\ \sigma(R(G)) &:= \sigma(G) \circ R^{-1}, & \delta(R(G)) &:= \delta(G) \circ R^{-1}, \end{aligned}$$

and to any pointed graph  $(G, v)$  by  $R(G, v) := (R(G), R(v))$ .

Operations are defined to manipulate graphs in a set-like fashion.

**Definition 3 (Consistency, Union, Intersection).** *Two graphs  $G$  and  $H$  are consistent when:*

- $E(G) \cup E(H)$  is a set of non-intersecting two-element sets,
- $\sigma(G)$  and  $\sigma(H)$  agree where they are both defined,
- $\delta(G)$  and  $\delta(H)$  agree where they are both defined.

In this case, the union  $G \cup H$  of  $G$  and  $H$  is defined by

$$\begin{aligned} V(G \cup H) &:= V(G) \cup V(H), & E(G \cup H) &:= E(G) \cup E(H), \\ \sigma(G \cup H) &:= \sigma(G) \cup \sigma(H), & \delta(G \cup H) &:= \delta(G) \cup \delta(H). \end{aligned}$$

The intersection  $G \cap H$  of  $G$  and  $H$  is always defined and given by

$$\begin{aligned} V(G \cap H) &:= V(G) \cap V(H), & E(G \cap H) &:= E(G) \cap E(H), \\ \sigma(G \cap H) &:= \sigma(G) \cap \sigma(H), & \delta(G \cap H) &:= \delta(G) \cap \delta(H). \end{aligned}$$

The empty graph with no vertex is designed  $\emptyset$ .

In this definition, the unions of functions seen as relations (that result in a non-functional relation in general) are here guaranteed to be functional by the consistency condition. The intersection of (partial) functions simply gives a (partial) function which is defined only on inputs on which both functions agree.

To describe the evolution of such graphs in the CGD framework, we first need to make precise the notion of locality, which is captured by how a graph is pruned to a local view: a disk. We consider the usual distance between two vertices in graphs, *i.e.*, the minimal number of edges of the paths between them, and denote by  $B_G(c, d)$  the set of vertices at distance at most  $d$  from  $c$  in the graph  $G$ .

**Definition 4 (Disk).** *Let  $G$  be a graph,  $c \in V(G)$  a vertex, and  $r$  a non-negative integer. The disk of radius  $r$  and center  $c$  is the pointed graph  $G_c^r = (H, c)$  with  $H$  given by*

$$\begin{aligned} V(H) &:= B_G(c, r+1), & \sigma(H) &:= \sigma(G) \upharpoonright B_G(c, r), \\ E(H) &:= \{\{u:i, v:j\} \in E(G) \mid \{u, v\} \cap B_G(c, r) \neq \emptyset\}, & \delta(H) &:= \delta(G) \upharpoonright E(H). \end{aligned}$$

We denote  $\mathcal{D}_{\Sigma, \Delta, \pi}^r$  the set  $\{G_c^r \mid G \in \mathcal{G}_{\Sigma, \Delta, \pi}, c \in V(G)\}$  of all disks of radius  $r$ , and  $\mathcal{D}_{\Sigma, \Delta, \pi}$  the set of all disks of any radius. When the disk notation is used with a set  $C$  of vertices as subscript, we mean

$$G_C^r := \bigcup_{c \in C} G_c^r. \quad (1)$$

The CGD dynamics relies on a local evolution describing how local views generates local outputs consistently.

**Definition 5 (Local Rule).** *A local rule of radius  $r$  is a function  $f : \mathcal{D}_{\Sigma, \Delta, \pi}^r \rightarrow \mathcal{G}_{\Sigma, \Delta, \pi}$  such that:*

1. *for any isomorphism  $R$ , there is another isomorphism  $R'$ , called the conjugate of  $R$ , with  $f \circ R = R' \circ f$ ,*
2. *for any family  $\{(H_i, v_i)\} \subset \mathcal{D}_{\Sigma, \Delta, \pi}^r$ ,  $\bigcap_i H_i = \emptyset \implies \bigcap_i f((H_i, v_i)) = \emptyset$ ,*
3. *there exists  $b$  such that for all  $D \in \mathcal{D}_{\Sigma, \Delta, \pi}^r$ ,  $|V(f(D))| \leq b$ ,*
4. *for any  $G \in \mathcal{G}_{\Sigma, \Delta, \pi}$ ,  $u, v \in V(G)$ ,  $f(G_v^r)$  and  $f(G_u^r)$  are consistent.*

In the second condition, note that a set of graphs have the empty graph as intersection iff their sets of vertices are disjoint. In the original work, functions respecting the two first conditions, the third condition, and the fourth condition are called respectively dynamics, bounded functions, and consistent functions. A local rule is therefore a bounded consistent dynamics.

The main result of [1] is the proof that *causal graph dynamics* are *localisable functions*, the concepts coming from the paper. We rely on this result in the following definition since we use the formal definition of the latter with the name of the former.

**Definition 6 (Causal Graph Dynamics (CGD)).** *A function  $F : \mathcal{G}_{\Sigma, \Delta, \pi} \rightarrow \mathcal{G}_{\Sigma, \Delta, \pi}$  is a causal graph dynamics (CGD) if there exists a radius  $r$  and a local rule  $f$  of radius  $r$  such that*

$$F(G) = \bigcup_{v \in V(G)} f(G_v^r). \tag{2}$$

### 2.3 Global Transformations & Kan Extensions

In category theory, Kan extensions are a construction allowing to extend a functor along another one in a universal way. In the context of this article, we restrict ourselves to the case of pointwise left Kan extensions involving only categories which are posets. In this case, their definition simplifies as follows.

**Definition 7 (Pointwise Left Kan Extension for Posets).** *Given three posets  $A$ ,  $B$  and  $C$ , and two monotonic functions  $i : A \rightarrow B$ ,  $f : A \rightarrow C$ , the function  $\Phi : B \rightarrow C$  given by*

$$\Phi(b) = \sup \{ f(a) \in C \mid a \in A \text{ s.t. } i(a) \preceq b \} \tag{3}$$

*is called the pointwise left Kan extension of  $f$  along  $i$  when it is well-defined (some suprema might not exist), and in which case it is necessarily monotonic.*

Global Transformations (GT) makes use of left Kan extensions to tackle the question of the synchronous deterministic local transformation of *arbitrary* kind of spaces. It is a categorical framework, but in the restricted case of posets, it works as follows. While  $B$  and  $C$  capture as posets the local-to-global relationship between the spatial elements to be handled (inputs and outputs respectively),  $A$  specifies a poset of local transformation rules. The monotonic functions  $i$  and  $f$  give respectively the left-hand-side and right-hand-side of the rules in  $A$ . Glancing at Eq. (3), the transformation mechanism works as follows. Consider an input spatial object  $b \in B$  to be transformed. The associated output  $\Phi(b)$  is obtained by gathering (thanks to the supremum in  $C$ ) all the right-hand-side  $f(a)$  of the rules  $a \in A$  with a left-hand-side  $i(a)$  occurring in  $b$ . The occurrence relationship is captured by the respective partial orders:  $i(a) \preceq b$  in  $B$  for the left-hand-side, and  $f(a) \preceq \Phi(b)$  in  $C$  for the right-hand-side.

The monotonicity of  $\Phi$  is the formal expression of a major property of a GT: if an input  $b$  is a subpart of an input  $b'$  (*i.e.*,  $b \preceq b'$  in  $B$ ), the output  $\Phi(b)$  has to occur as a subpart of the output  $\Phi(b')$  (*i.e.*,  $\Phi(b) \preceq \Phi(b')$  in  $C$ ). This property gives to the orders of  $B$  and  $C$  a particular semantics for GT which will play an important role in the present work.

*Remark 1.* Elements of  $B$  are understood as information about the input. So, when  $b \preceq b'$ ,  $b'$  provides a richer information than  $b$  about the input that  $\Phi$  uses to produce output  $\Phi(b')$ , itself richer than output  $\Phi(b)$ . However,  $\Phi$  cannot deduce the falsity of a property about the input from the fact this property is not included in  $b$ ; otherwise the output  $\Phi(b)$  might be incompatible with  $\Phi(b')$ .

At the categorical level, the whole GT formalism relies on the key ingredient that the collection  $A$  of rules is also a category. Arrows in  $A$  are called *rule inclusions*. They guide the construction of the output and allow overlapping rules to be applied all together avoiding the well-known issue of concurrent rules application [5].

In cases where  $\Phi$  captures the evolution function of a (discrete time) dynamical system (so particularly for the present work where we want to compare  $\Phi$  to a CGD), we consider the input and output categories/posets  $B$  and  $C$  to be the same category/poset, making  $\Phi$  an endo-functor/function.

### 3 Unifying Causal Graph Dynamics and Kan Extensions

The starting point of our study is that Eq. (2) in the definition of CGD has almost the same form as Eq. (3). Indeed, if we take Eq. (3) and set  $A = \mathcal{D}_{\Sigma, \Delta, \pi}^r$ ,  $B = C = \mathcal{G}_{\Sigma, \Delta, \pi}$ , the function  $i$  to be the projection function from discs to graphs that drops their centers (*i.e.*,  $i((H, c)) = H$ ) and the function  $f$  to be the local function from discs to graphs, we obtain an equation for  $\Phi$  of the form

$$\Phi(G) = \sup \{ f(D) \in \mathcal{G}_{\Sigma, \Delta, \pi} \mid D \in \mathcal{D}_{\Sigma, \Delta, \pi}^r \text{ s.t. } i(D) \preceq G \} \quad (4)$$

which is close to the equation Eq. (2) rewritten

$$F(G) = \bigcup \{ f(D) \in \mathcal{G}_{\Sigma, \Delta, \pi} \mid D = G_v^r, v \in V(G) \}.$$

This brings many questions. What is the partial order involved in Eq. (4)? Is the union of Eq. (2) given by the suprema of this partial order? Is it the case that  $i(D) \preceq G$  implies  $D = G_v^r$  for some  $v$  in this order? Are  $f$  and  $F$  of Eq. (2) monotonic functions for this order? We tackle these questions in the following sections.

#### 3.1 The Underlying Partial Order

Considering the two first questions, there is a partial order which is forced on us. Indeed, we need this partial order to imply that suprema are unions of graphs. But partial orders can be defined from their binary suprema since  $A \preceq B \iff \sup \{A, B\} = B$ . Let us give explicitly the partial order, since it is very natural, and prove afterward that it is the one given by the previous procedure.

**Definition 8 (Subgraph).** *Given two graphs  $G$  and  $H$ ,  $G$  is a subgraph of  $H$ , denoted  $G \subseteq H$ , when*

$$V(G) \subseteq V(H) \wedge E(G) \subseteq E(H) \wedge \sigma(G) \subseteq \sigma(H) \wedge \delta(G) \subseteq \delta(H).$$

This defines a partial order  $-\subseteq-$  on  $\mathcal{G}_{\Sigma,\Delta,\pi}$  called the subgraph order. The subgraph order is extended to pointed graphs by

$$(G, v) \subseteq (H, u) : \iff G \subseteq H \wedge v = u.$$

In this definition, the relational condition  $\sigma(G) \subseteq \sigma(H)$ , where these two functions are taken as sets of input-output pairs, means that  $\sigma(G)(v)$  is either undefined or equal to  $\sigma(H)(v)$ , for any vertex  $v \in V(G)$ . The same holds for the condition  $\delta(G) \subseteq \delta(H)$ .

Let us now state that the subgraph order has the correct relation with unions of graphs. It similarly encodes consistency and intersections of pairs of graphs.

**Proposition 1.** *Two graphs  $G$  and  $H$  are consistent precisely when they admit an upper bound in  $(\mathcal{G}_{\Sigma,\Delta,\pi}, \subseteq)$ . The union of  $G$  and  $H$  is exactly their supremum (least upper bound) in  $(\mathcal{G}_{\Sigma,\Delta,\pi}, \subseteq)$ . The intersection of  $G$  and  $H$  is exactly their infimum (greatest lower bound) in  $(\mathcal{G}_{\Sigma,\Delta,\pi}, \subseteq)$ .*

The two first questions being answered positively, let us rewrite Eq. (4) as

$$\Phi(G) = \bigcup \{ f(D) \mid D \in \mathcal{D}_{\Sigma,\Delta,\pi}^r \text{ s.t. } i(D) \subseteq G \}. \quad (5)$$

### 3.2 Comparing Disks and Subgraphs

Let us embark on the third question: is it the case that  $i(D) \subseteq G$  implies  $D = G_v^r$  for some  $v$ ? Making the long story short, the answer is no. But it is crucial to understand precisely why. Fix some vertex  $v \in V(G)$ . Clearly, in Eq. (2), the only considered disk centered on  $v$  is  $G_v^r$ . Let us determine now what are exactly the disks centered on  $v$  involved in Eq. (5), that is, the set  $I_v := \{(H, v) \in \mathcal{D}_{\Sigma,\Delta,\pi}^r \mid H \subseteq G\}$ . Firstly,  $G_v^r$  is one of them of course.

**Lemma 1.** *For any vertex  $v \in V(G)$ ,  $G_v^r \in I_v$ .*

The concern is that  $G_v^r$  is generally not the only one disk in  $I_v$  as expected by Eq. (2). However,  $G_v^r$  is the maximal one in the following sense.

**Lemma 2.** *For any vertex  $v \in V(G)$ , consider the disk  $G_v^r = (H, v)$ . Then for any disk  $(H', v) \in I_v$ , we have  $H' \subseteq H$ .*

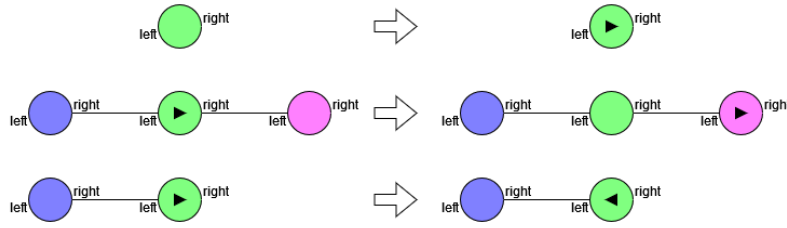
In some sense, the converse of the previous proposition holds: it is roughly enough to be smaller than  $G_v^r$  to be a disk of  $I_v$ , as characterized by the following two propositions.

**Lemma 3.** *The set of disks is the set of pointed connected finite graphs.*

**Proposition 2.** *For any vertex  $v \in V(G)$ ,  $I_v$  is a principal downward closed set in the poset of graphs restricted to connected finite graphs containing  $v$ .*

We now know that the union of Eq. (5) receives a bigger set of local outputs to merge than the union of Eq. (2). But we cannot conclude anything yet. Indeed, it might be the case that all additional local outputs do not contribute anything more. This is in particular the case if disks  $D \in I_v$  are such that  $f(D) \subseteq f(G_v^r)$ . This is related to the fourth and last question.





**Fig. 1.** Moving particle CGD - non-monotonic behavior. Each row represents an example of evolution with a graph  $G$  on the left and  $F(G)$  on the right. Colors correspond to vertex names.

### 3.3 Monotonic and General Causal Graph Dynamics

The last remark invites us to consider the case where the CGD is monotonic. We deal with the general case afterward.

**Monotonic CGD as Kan Extensions.** As just evoked, things seem to go well if the local rule  $f$  happens to be monotonic. All the ingredients have been already given and the proposition can be made formal straightforwardly.

**Proposition 3.** *Let  $F : \mathcal{G}_{\Sigma, \Delta, \pi} \rightarrow \mathcal{G}_{\Sigma, \Delta, \pi}$  be a CGD with local rule  $f : \mathcal{D}_{\Sigma, \Delta, \pi}^r \rightarrow \mathcal{G}_{\Sigma, \Delta, \pi}$  of radius  $r$ . If  $f$  is monotonic, then  $F$  is the pointwise left Kan extension of  $f$  along  $i : \mathcal{D}_{\Sigma, \Delta, \pi}^r \rightarrow \mathcal{G}_{\Sigma, \Delta, \pi}$ , the projection of discs to graphs.*

*Proof.* The proposition is equivalent to show that  $F(G) = \Phi(G)$  for all  $G$ . Summarizing our journey up to here, we now know that Eq. (5) and Eq. (2) are similar except that the former iterates over the set of disks  $I = \bigcup_{v \in V(G)} I_v$  while the latter iterates over  $J = \{G_v^r \mid v \in V(G)\}$ , with  $J \subseteq I$  by Prop. 1. We get  $F(G) \subseteq \Phi(G)$ . Moreover, by Prop. 2, for any  $D \in I_v$ , for any  $v$ ,  $D \subseteq G_v^r$ , and by monotonicity of  $f$ ,  $f(D) \subseteq f(G_v^r)$ . So  $f(D) \subseteq F(G)$  for all  $D$  and  $\Phi(G) \subseteq F(G)$ , leading to the expected equality.  $\square$

The class of CGD having such a monotonic local rule is easily characterizable: they correspond to CGD that are monotonic themselves as stated by the following proposition.

**Proposition 4.** *A CGD  $F$  is monotonic iff  $F$  admits a monotonic local rule.*

**Corollary 1.** *A CGD is monotonic iff it is a left Kan extension.*

**The Non-Monotonic Case.** The previous result brings us close to our initial goal: encoding any CGD as a GT. The job would be considered done only if there is no non-monotonic CGD, or if those CGD are degenerate cases. However, it is clearly not the case and most of the examples in the literature are of this kind, as we can see with the following example, inspired from [3].

Consider for example the modeling of a particle going left and right on a linear graph by bouncing at the extremities. The linear structure is coded using two ports “left” and “right” on each vertex, while the particle is represented with the presence of a label on some vertex, with two possible values indicating its direction. See Fig. 1 for illustrations of such graphs. The dynamics of the particle is captured by  $F$  as follows. Suppose that the particle is located at some vertex  $v$  (in green in Fig. 1), and wants to go to the right. If there is an outgoing edge to the right to an unlabeled vertex  $u$  (in pink in Fig. 1), the label representing the particle is moved from  $v$  to  $u$  (second row of Fig. 1). If there is no outgoing edge to the right (the “right” port of  $v$  is free), it bounces by becoming a left-going particle (third row of Fig. 1).  $F$  works symmetrically for a left-going particle.

The behavior of  $F$  is non-monotonic since the latter situation is a sub-graph of the former, while the particle behaviors in the two cases are clearly incompatible. On Fig. 1, the right hand sides of the second and third rows are comparable by inclusion, but the left hand sides are not. This non-monotonicity involves a missing edge but missing labels may induce non-monotonicity as well. Suppose for the sake of the argument that  $F$  generates a right-going particle at any unlabeled isolated vertex (first row of Fig. 1). The unlabeled one-vertex graph is clearly a subgraph of any other one where the same vertex is labeled by a right-going particle and has some unlabeled right neighbor. In the former case, the dynamics puts a label on the vertex, while it removes it on the latter case. The new configurations are no longer comparable. See the first and second rows of Fig. 1 for an illustration.

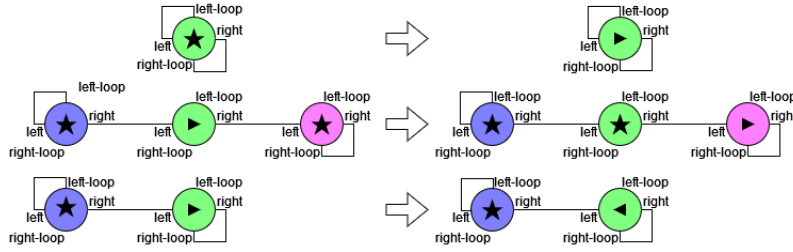
**Proposition 5.** *CGD are not necessarily monotonic.*

From Corollary 1, we conclude that all non-monotonic CGD are not left Kan extensions as we have developed so far, *i.e.*, based on the subgraph relationship of Def 8. Analyzing the particle CGD in the light of Remark 1 tells us why. Indeed, the subgraph ordering is able to compare a place without any right neighbor with a place with some (left-hand-side of rows 3 and 2 in Fig. 1 for instance). Following Remark 1, in the GT setting, the former situation has less information than the latter: in the former, there is no clue whether the place has a neighbor or not; the dynamics should not be able to specify any behavior for a particle at that place. But clearly, for the corresponding CGD, both situations are totally different: the former is an extremity while the latter is not, and the dynamics specifies two different behaviors accordingly for a particle at that place.

## 4 Universality of Monotonic Causal Graph Dynamics

We have proven that the set of all CGD is strictly bigger than the set of monotonic ones. However, we prove now that it is not more expressive. By this, we mean that we can simulate any CGD by a monotonic one, *i.e.*, monotonic CGD are universal among general CGD.

More precisely, given a *general* CGD  $F$ , a monotonic simulation of  $F$  consists of encoding, call it  $\omega(G)$ , of each graph  $G$ , and a *monotonic* CGD  $F'$  such that



**Fig. 2.** Moving particle CGD - monotonic behavior. Compared to Fig. 1, each vertex has two additional ports and unlabeled vertices are now labeled by  $\star$ .

whenever  $F(G) = H$  on the *general* side,  $F'(\omega(G)) = \omega(H)$  on the *monotonic* side. Substituting  $H$  in the latter equation using the former equation, we get  $F'(\omega(G)) = \omega(F(G))$ , the exact property of the expected simulation: for any  $F$ , we want some  $\omega$  and  $F'$  such that  $F' \circ \omega = \omega \circ F$ .

#### 4.1 Key Ideas of the Simulation

In this section, we aim at introducing the key elements of the simulation informally and by the mean of the moving particle example.

**Encoding the Original Graphs: The Moving Particle Case.** Let us design a monotonic simulation of the particle dynamics. The original dynamics can be made monotonic by replacing the two missing edges at the extremities by easily identifiable loopback edges, making incomparable the two originally comparable situations. For such loopback edges to exist, we need an additional port for each original port, in our case say “left-loop” and “right-loop” for instance. For the case of non-monotonicity with labels, vertices where there is no particle (so originally unlabeled) are marked with a special label, say  $\star$ . Fig. 2 depicts the same evolutions as Fig. 1 after those transformations.

This will be the exact role of the encoding function  $\omega$ : the key idea to design a monotonic simulation is to make uncomparable the initially comparable situations. Any missing entity (edge or label) composing an original graph  $G$  needs to be replaced by a special entity (loopback edge or  $\star$  respectively) in  $\omega(G)$  indicating it was originally missing. At the end of the day, for any  $G \subsetneq H$ , we get  $\omega(G)$  and  $\omega(H)$  no longer comparable.

**An Extended Universe of Graphs.** All that remains is to design  $F'$  such that  $F' \circ \omega = \omega \circ F$ . It is important to note however that the universe of graphs targeted by  $\omega$ , which is the domain of  $F'$ , is by design much larger than the original universe. Indeed, each vertex has now a doubled number of ports and the additional label  $\star$  is available. So to be completely done with the task, we



**Fig. 3.** Different classes of graphs: from left to right, a total graph, a coherent partial graph, an incoherent graph.

need  $F'$  to be able to work not only with graphs generated by the encoding, but also with all the other graphs.

Let us classify the various cases. Firstly, notice that the monotonic counterpart  $\omega(G)$  of any graph  $G$  is “total” in the following sense: all vertices have labels, all original ports have edges, and all edges have labels. However, there also exists partial graphs with free ports and unlabeled vertices in this monotonic universe of graphs. Secondly,  $\omega(G)$  uses the additional ports strictly for the encoding of missing edges with loopback edges. But, there are also graphs making arbitrary use of those ports and which are not “coherent” with respect to the encoding. Fig. 3 illustrates the three identified classes: the middle graph is a coherent subgraph of the total graph on the right, and of an incoherent graph on the left. Notice that incoherent edges can always be dropped away to get the largest coherent subgraph of any graph. This is the case on the figure.

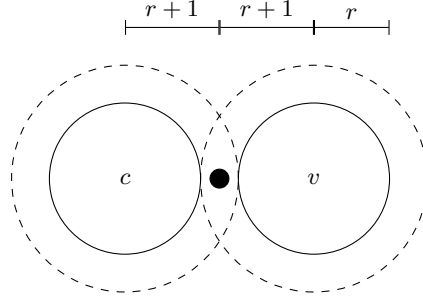
**Disks in the Extended Universe.** In order to design a local rule  $f'$  of the monotonic CGD  $F'$ , we need to handle disks after encoding. The universe of graphs being bigger than initially, this also holds for disks, and the behavior of  $f'$  will depend on the class of the disk.

Let us first identify the monotonic counterpart of the original disks. They are called “total” disks and correspond to disks of some  $\omega(G)$ . For those disks, the injectivity of  $\omega$  allows  $f'$  to simply retrieve the corresponding disk of the original graph  $G$  and invoke the original local rule  $f$  on it.

An arbitrary disk  $D$  may not be “total” and exhibit some partiality (free ports, unlabeled vertices or edges). Since  $f'$  is required to be monotonic, it needs to output a subgraph of the original local rule output. More precisely, we need to have  $f'(D) \subseteq f'(D')$  for any total disk  $D'$  with  $D \subseteq D'$ . The easiest way to do that is by outputting the empty graph:  $f'(D) = \emptyset$ . (This solution corresponds to the so-called *coarse extension* proposed for CA in [4]; *finer extensions* are also considered there.)

Last, but not least, an arbitrary disk  $D$  may make an incoherent use of the additional ports with respect to  $\omega$ . In that case, all incoherent information may be ignored by considering the largest coherent subdisk  $D' \subseteq D$ . The behavior of  $f'$  on  $D$  is then aligned with its behavior on  $D'$ :  $f'(D) = f'(D')$ .

**A Larger Radius.** The last parameter of  $f'$  to tune is its radius. It will of course depend on the radius of the local rule of  $F$ . But it is worth noting that we are actually trying to build more information locally than the original local



**Fig. 4.** We need to determine for each missing entity attached to a vertex or edge of  $f(G_c^r)$ , whether all other  $f(G_v^r)$  agree to consider this entity as missing. The property 2 of local rules (Def. 5) tells us that only disks  $G_v^r$  that intersect  $G_c^r$  need to be checked. The furthest such  $v$  are at distance  $2r + 2$ . From there, we need to ask for radius  $r$  (so radius  $r' = 3r + 2$  from  $c$ ) to include the entirety of  $G_v^r$  (including its border at  $r + 1$  from  $v$ , and therefore  $3r + 3$  at most from  $c$ ).

rule was trying to. Indeed, given a vertex of the original global output, many local rule applications may contribute concurrently to its definition. All of these contributions are consistent with each other of course, but it is possible for some local outputs to indicate some features of that vertex (label, edges) while others do not. If even one of them puts such a label for instance, then there is a label in the global output. It is only if none of them put a label that the global output will not have any label on it. The same holds similarly for ports: a port is free in the global output if it is so on all local outputs. Because the monotonic counterpart needs to specify locally if none of the local rule applications puts such a feature to a vertex, the radius of the monotonic local rule needs to be big enough to include all those local rule input disks. It turns out that the required radius for  $f'$  is  $r' = 3r + 2$ , if  $r$  is the radius of the original local rule  $f$ , as explained in Fig. 4.

## 4.2 Formal Definition of the Simulation

Now that we have all the components of the solution, let us make them precise.

**The Encoding Function  $\omega$ .** The encoding function aims at embedding the graphs of  $\mathcal{G}_{\Sigma, \Delta, \pi}$  in a universe where ports are doubled and labels are extended with an extra symbol. Set  $\Sigma' := \Sigma \uplus \{\star\}$ ,  $\Delta' := \Delta \uplus \{\star\}$ , and  $\pi' = \pi \times \{0, 1\}$ . Ports  $(a, 0) \in \pi'$  are considered to be semantically the same as their counterpart  $a \in \pi$  while ports  $(a, 1)$  are their for loopback edges. For simplicity, let us define the short-hands  $\mathcal{G}$  and  $\mathcal{G}'$  for  $\mathcal{G}_{\Sigma, \Delta, \pi}$  and  $\mathcal{G}_{\Sigma', \Delta', \pi'}$  respectively for the remainder of this section. Since we need to complete partial functions, we introduce the following notation: for any partial function  $f : X \rightharpoonup Y$  and total function  $z :$

$X \rightarrow Z$ , we define the total function  $f!z : X \rightarrow Y \cup Z$

$$(f!z)(x) = \begin{cases} f(x) & \text{if } f(x) \text{ is defined,} \\ z(x) & \text{otherwise.} \end{cases}$$

For simplicity, we write  $f!*$  for  $f!(x \mapsto *)$  where  $x$  runs over  $X$ . This allows us to define the encoding  $\omega(G) \in \mathcal{G}'$  of any  $G \in \mathcal{G}$  as follows.

$$\begin{aligned} V(\omega(G)) &:= V(G) \\ \sigma(\omega(G)) &:= \sigma(G)!* \\ E(\omega(G)) &:= \{ \{u:(i,0), \text{trg}_G^*(u:i)\} \mid u:i \in V(G):\pi \} \\ \delta(\omega(G))(\{u:(i,a), v:(j,b)\}) &:= \begin{cases} (\delta(G)!*)(\{u:0, v:j\}) & \text{if } b = 0 \\ * & \text{if } u:i = v:j \end{cases} \end{aligned}$$

The definition of  $E(\omega(G))$  is written to make clear that all original ports are indeed occupied by an edge. The total function  $\text{trg}_G^* : V(G) \times \pi \rightarrow V(G) \times \pi'$  is defined based on the partial function  $\text{trg}_G : V(G) \times \pi \rightarrow V(G) \times \pi$  as follows.

$$\begin{aligned} \text{trg}_G(u:i) &:= v:j \text{ iff } \{u:i, v:j\} \in E(G) \\ \text{trg}_G^* &:= (t_0 \circ \text{trg}_G)!t_1 \text{ where } t_m(u:i) := u:(i, m) \end{aligned}$$

The definition of  $\delta(\omega(G))$  deals with original edges for the first case, and with loopback edges for the second, which are the only two possibilities with respect to  $\text{trg}_G^*$ .

**The Disk Encoding Function  $\omega_r$ .** Let us now define the shorthands  $\mathcal{D}_r$  and  $\mathcal{D}'_r$  for  $\mathcal{D}_{\Sigma, \Delta, \pi}^r$  and  $\mathcal{D}'_{\Sigma', \Delta', \pi'}$  for any radius  $r$ . We put the radius as a subscript to make readable the four combinations  $\mathcal{D}_r$ ,  $\mathcal{D}'_r$ ,  $\mathcal{D}_{r'}$ , and  $\mathcal{D}'_{r'}$ . The function  $\omega_r : \mathcal{D}_r \rightarrow \mathcal{D}'_r$  aims at encoding original disks as total disks. It is defined as

$$\omega_r((D, v)) = \omega(D)_v^r. \quad (6)$$

The reason we apply  $(-)_v^r : \mathcal{G}' \rightarrow \mathcal{D}'_r$  on the raw result of  $\omega$  is that the behavior of  $\omega$  (adding loopback edges on each unused port and labels on unlabeled vertices and edges) is not desired at distance  $r + 1$  for the result to be a disk of radius  $r$ . More conceptually,  $\omega_r$  is the unique function having the following commutation property, which expresses that  $\omega_r$  is the disk counterpart of  $\omega$ .

**Proposition 6.** *For any graph  $G \in \mathcal{G}$ ,  $\omega_r(G_v^r) = \omega(G)_v^r$ .*

**Coherent Subgraph Function  $\text{coh}$ .** As discussed informally, some graphs in  $\mathcal{G}'$  may be ill-formed from the perspective of the encoding by using doubled ports

arbitrarily. We define the function  $\text{coh} : \mathcal{G}' \rightarrow \mathcal{G}'$  that removes the incoherencies.

$$\begin{aligned} V(\text{coh}(G')) &:= V(G') \\ \sigma(\text{coh}(G')) &:= \sigma(G') \\ E(\text{coh}(G')) &:= \{\{u:(i,0),v:(j,b)\} \in E(G') \mid \\ &\quad b=1 \Rightarrow (v:j=u:i \wedge \delta(G')(\{u:(i,0),u:(i,1)\}) = \star)\} \\ \delta(\text{coh}(G')) &:= \delta(G') \upharpoonright E(\text{coh}(G')) \end{aligned}$$

Since  $\text{coh}$  only removes incoherent edges, and since they are stable by inclusion, we trivially have that

**Lemma 4.** *coh is monotonic.*

**The Monotonic Local Rule.** The monotonic local rule  $f'$  is defined in two stages. The first stage is to take a disk in  $\mathcal{D}'_{r'}$  and transform it into something that the original local rule can work with. To do this, we remove any incoherencies with the function  $\text{coh}$ , and if the result is total, we restrict it to the correct radius and retrieve its counterpart in the original universe (which is possible because  $\omega_r$  is injective). The original local function can be called on this counterpart. As discussed before, if the coherent sub-disk is not total, we simply output an empty graph. The signature of this function is  $\phi : \mathcal{D}'_{r'} \rightarrow \mathcal{G}$ .

$$\phi(D', c) := \begin{cases} \emptyset & \text{if } \text{coh}(D') \notin \text{Im}(\omega_{r'}) \\ f(\omega_r^{-1}(\text{coh}(D')^c)) & \text{otherwise} \end{cases} \quad (7)$$

Turning this result in  $\mathcal{G}$  into its counterpart in  $\mathcal{G}'$  is more complicated than simply invoking  $\omega$  since one has to check that missing entities are really missing, as discussed in Fig. 4. This is the purpose of the second stage leading to the definition of the monotonous local rule  $f' : \mathcal{D}'_{r'} \rightarrow \mathcal{G}'$  which, thanks to its radius  $r' = 3r + 2$ , is able to inspect all  $r$ -disks at distance  $2r + 2$  from  $c$ .

$$f'((D', c)) = \omega \left( \bigcup_{v \in \mathcal{D}'_c^{2r+1}} \phi(D', v) \right)_{\phi(D', c)}^0 \quad (8)$$

In this equation, the union of all local results that can contribute to entities attached to  $\phi(D', c)$  is built. In this way, they are all given a chance to say if what was missing in  $\phi(D', c)$  is actually missing. Finally  $\omega$  is applied and its result restricted to the sole vertices of  $\phi(D', c)$  and their edges, using the notation  $(-)_X^0$  of Eq. (1). Those vertices and edges are the only one for which all the possibly contributing disks have been inspected.

**Proposition 7.** *f' is a monotonic local rule.*

**The Monotonic Simulation.** We finally have the wanted CGD  $F'$  of local rule  $f'$ :

$$F'(G') := \bigcup_{v \in V(G')} f'(G'_{v'}).$$

It remains then to show that  $F'$  is indeed a monotonic simulation of  $F$ , which is achieved with the two next propositions.

**Proposition 8.**  $F'$  is monotonic.

*Proof.* Since  $f'$  is monotonic by Prop. 7,  $F'$  is monotonic as well by Prop. 4.  $\square$

**Proposition 9.**  $F'$  simulates  $F$  via the  $\omega$  encoding, i.e.,  $F' \circ \omega = \omega \circ F$ .

*Proof.* Suppose  $G' = \omega(G)$  for some  $G \in \mathcal{G}$ . Since all involved disks in  $F'(\omega(G))$  are total and thanks to Prop. 6, the expression of  $F'(\omega(G))$  simplifies drastically.

$$F'(\omega(G)) = \bigcup_{c \in V(G)} \omega \left( \bigcup_{v \in G_c^{2r+1}} f(G_v^r) \right)_{f(G_c^r)}^0. \quad (9)$$

Clearly,  $F'(\omega(G))$  does not exhibit any incoherencies and  $\omega(F(G))$  is total, so it is enough to show that  $\omega(F(G)) \subseteq F'(\omega(G))$  to have the equality. This can be done entity by entity. Take a vertex  $u \in \omega(F(G))$ . It comes from some  $f(G_c^r)$ . So it belongs to the inner union of Eq. (9). It is preserved by  $\omega$  then by  $(-)^0_{f(G_c^r)}$ , so it belongs to  $F'(\omega(G))$ . Consider now its label  $\ell := \delta(\omega(F(G)))(u)$ . If  $\ell \neq \star$ , there is some  $v \in G_c^{2r+1}$  with  $\ell = \delta(f(G_v^r))(u)$ . So the inner union of Eq. (9) labels  $u$  by  $\ell$ . The label is preserved by  $\omega$  then by  $(-)^0_{f(G_c^r)}$  since  $u \in f(G_c^r)$ . So  $\delta(F'(\omega(G)))(u) = \ell$  as well. If  $\ell = \star$ , this means that none of the  $f(G_v^r)$  put a label on  $u$ , so  $u$  is unlabeled in the inner union of Eq. (9). By definition,  $\omega$  completes the labeling by  $\star$  at  $u$ , and  $(-)^0_{f(G_c^r)}$  preserves this label. So  $\delta(F'(\omega(G)))(u) = \star$  as well. The proof continues similarly for edges and their labels, with an additional care for dealing with loopback edges.  $\square$

## 5 Conclusion

In this article, we planned to compare CGD and GT frameworks. The very particular route we have chosen for this task led us to identify the class of Monotonic CGD which are both CGD and GT, and happen to be universal among all CGD.

This work was guided by the formal similarities between the two frameworks leading to a list of four questions. The chosen strategy to cope with these questions was to tackle the first one “what is the order?” with the goal of answering positively to the second question “is the union of CGD the supremum of this order?”. This journey led us to identify the subgraph order to structure the set of port graphs instead of considering them independent (more precisely related by the disks only) as the original framework does. This pushes forward the idea of using graph inclusion to express gain of information as proposed by the GT framework, opening a new direction when designing CGD by respecting the order with monotonicity.

One interesting thing to note is that a slight adaptation of the original definition of port graphs allows to represent general CGD as Kan extensions without



any encoding. Indeed, the encoding considered in this article leads to a universe of graphs where most of them are ill-formed. By adding explicitly to graphs additional features to represent the positive information that some port is not occupied or some label is missing, it is possible to get a universe where all objects make sense.

An alternative route to answer the four-question list is possible by tackling the first question with the goal of answering the third one (almost) positively, thus falsifying the second one. An order closely related to this route is the “induced subgraph” order stating that  $G$  is lower than  $H$  if  $G = H_{V(G)}^0$ . This order is stronger than the usual subgraph order but is also very interesting. Fewer graphs are “inducedly consistent” and one may ask if the subclass of CGD definable with this stronger notion of consistency is universal. This might give an idea on whether the result of this article is isolated or, on the contrary, if it follows a common pattern shared with many instances.

Finally, let us not forget the next steps sketched in the introduction, in particular the quotienting of vertex names. In this paper, the notions of isomorphism and renaming-invariance of CGD play no real role. Because GT use a categorical language, this renaming-invariance can be handled by quotienting the objects of the poset, but not the arrows between the objects. This means that there would be now typically many arrows between two objects that the GT framework is able to cope with. Indeed, the formalism of GT was designed to work with objects like abstract graphs where no notion of names or positioning exist, but only their (possible multiple) occurrences in each other. As similar line of thought was already explored for cellular automata in [4]. Indeed, cellular automata are typically defined with a global positioning system, but many studies actually only care about the relative positioning of the information. A similar treatment for CGD is for futur work.

## References

1. Arrighi, P., Dowek, G.: Causal graph dynamics. In: Czumaj, A., Mehlhorn, K., Pitts, A.M., Wattenhofer, R. (eds.) *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II. Lecture Notes in Computer Science*, vol. 7392, pp. 54–66. Springer (2012). [https://doi.org/10.1007/978-3-642-31585-5\\_9](https://doi.org/10.1007/978-3-642-31585-5_9), [https://doi.org/10.1007/978-3-642-31585-5\\_9](https://doi.org/10.1007/978-3-642-31585-5_9)
2. Arrighi, P., Martiel, S.: Quantum causal graph dynamics. *Physical Review D* **96**(2), 024026 (2017)
3. Arrighi, P., Martiel, S., Perdrix, S.: Reversible causal graph dynamics: invertibility, block representation, vertex-preservation. *Nat. Comput.* **19**(1), 157–178 (2020). <https://doi.org/10.1007/S11047-019-09768-0>, <https://doi.org/10.1007/s11047-019-09768-0>
4. Fernandez, A., Maignan, L., Spicher, A.: Cellular automata and kan extensions. *Nat. Comput.* **22**(3), 493–507 (2023). <https://doi.org/10.1007/S11047-022-09931-0>, <https://doi.org/10.1007/s11047-022-09931-0>
5. Maignan, L., Spicher, A.: Global graph transformations. In: Plump, D. (ed.) *Proceedings of the 6th International Workshop on Graph Computation Models co-*

located with the 8th International Conference on Graph Transformation (ICGT 2015) part of the Software Technologies: Applications and Foundations (STAF 2015) federation of conferences, L'Aquila, Italy, July 20, 2015. CEUR Workshop Proceedings, vol. 1403, pp. 34–49. CEUR-WS.org (2015), <http://ceur-ws.org/Vol-1403/paper4.pdf>

## A Proofs for Section 3 (Unifying Causal Graph Dynamics and Kan Extensions)

**Proposition 1.** *Two graphs  $G$  and  $H$  are consistent precisely when they admit an upper bound in  $(\mathcal{G}_{\Sigma, \Delta, \pi}, \subseteq)$ . The union of  $G$  and  $H$  is exactly their supremum (least upper bound) in  $(\mathcal{G}_{\Sigma, \Delta, \pi}, \subseteq)$ . The intersection of  $G$  and  $H$  is exactly their infimum (greatest lower bound) in  $(\mathcal{G}_{\Sigma, \Delta, \pi}, \subseteq)$ .*

*Proof.* Admitting an upper bound in  $\mathcal{G}_{\Sigma, \Delta, \pi}$  means there is  $K \in \mathcal{G}_{\Sigma, \Delta, \pi}$  such that  $G \subseteq K$  and  $H \subseteq K$ . Since the subgraph order is defined componentwise, we consider the union of  $G$  and  $H$  as quadruplets of sets:

$$(V(G) \cup V(H), E(G) \cup E(H), \sigma(G) \cup \sigma(H), \delta(G) \cup \delta(H)),$$

which always exists and is the least upper bound in the poset of quadruplets of sets with the natural order. For this object to be a graph, it is enough to check  $E(G) \cup E(H)$  is a non-intersecting two-element set, and that  $\sigma(G)$  and  $\sigma(H)$  (resp.  $\delta(G)$  and  $\delta(H)$ ) coincide on their common domain. This is the case when  $G$  and  $H$  admit an upper bound as required in the definition. Conversely, when the condition holds, the union is itself an upper bound.

For intersection, consider

$$(V(G) \cap V(H), E(G) \cap E(H), \sigma(G) \cap \sigma(H), \delta(G) \cap \delta(H))$$

which is clearly the greatest lower bound and is always a graph.  $\square$

**Lemma 1.** *For any vertex  $v \in V(G)$ ,  $G_v^r \in I_v$ .*

*Proof.* In the Def. 4, the graph component  $H$  of the pointed graph  $G_v^r$  is explicitly defined by taking a subset for each of the four components of  $G$ , as required in Def. 8 of subgraph.  $\square$

**Lemma 2.** *For any vertex  $v \in V(G)$ , consider the disk  $G_v^r = (H, v)$ . Then for any disk  $(H', v) \in I_v$ , we have  $H' \subseteq H$ .*

*Proof.* By Def. 8 of the subgraph order, we need to prove four inclusions. For the inclusion of vertices, consider an arbitrary vertex  $w \in V(H')$ . By definition of disks of radius  $r$ , there is a path in  $H'$  from  $v$  to  $w$  of length at most  $r+1$ . But since  $H' \subseteq G$ , we have  $w \in V(G)$  and this path itself is also in  $G$ . So  $w \in V(G)$  respects the defining property of the set  $V(G_v^r)$  and therefore belongs to it. The three other inclusions ( $E(H') \subseteq E(H)$ ,  $\sigma(H') \subseteq \sigma(H)$  and  $\delta(H') \subseteq \delta(H)$ ) are proved similarly, by using the definition of disks, then the fact  $H' \subseteq G$ , and finally the definition of  $G_v^r$ .  $\square$

**Lemma 3.** *The set of disks is the set of pointed connected finite graphs.*

*Proof.* Indeed, for any disk  $(H, v) \in \mathcal{D}_{\Sigma, \Delta, \pi}$ ,  $H$  is connected since all vertices are connected to  $v$ , and  $H$  is finite since all vertices have at most  $|\pi|$  neighbors, so a rough bound is  $|\pi|^r$  with  $r$  the radius of the disk. Conversely, for any pointed connected finite graphs  $(G, v)$ , we have  $(G, v) = G_v^r$  for any  $r \geq |V(G)| - 1$ , the length of the longest possible path.  $\square$

**Proposition 2.** *For any vertex  $v \in V(G)$ ,  $I_v$  is a principal downward closed set in the poset of graphs restricted to connected finite graphs containing  $v$ .*

*Proof.* Indeed, consider any disk  $(H, v)$  such that  $H \subseteq G$ . Now, take  $H'$  a connected graph containing  $v$  and such that  $H' \subseteq H$ . Since  $H$  is finite, so is  $H'$ . By Proposition 3,  $(H', v)$  is also a disk. By transitivity,  $H' \subseteq H \subseteq G$ . This proves that we have a downward closed set. This is moreover a principal one because of Propositions 1 and 2.  $\square$

**Proposition 4.** *A CGD  $F$  is monotonic iff  $F$  admits a monotonic local rule.*

*Proof.* If  $F$  has a monotonic local rule  $f$ ,  $F$  is a left Kan extension by Prop. 3 and is monotonic as recalled in Sec. 2.3.

Conversely, suppose  $F$  monotonic. By Def. 6 of CGD, there is a local rule  $f'$ , not necessarily monotonic, of radius  $r$  generating  $F$ . Consider  $f$  defined by  $f((H, c)) = F(H)$  for any disk  $(H, c)$ .  $f$  is monotonic by monotonicity of  $F$ .  $f$  is a local rule, which is checked easily using that  $f'$  is itself a local rule and that  $f((H, c)) = \bigcup_{v \in H} f'(H_v^r)$ .  $f$  generates  $F$ . For any graph  $G$ :

$$\bigcup_{c \in V(G)} f(G_c^r) = \bigcup_{c \in V(G)} \bigcup_{v \in G_c^r} f'(i(G_c^r)_v^r).$$

But  $f'(i(G_c^r)_v^r) \subseteq F(i(G_c^r)_v^r) \subseteq F(G)$ , the last inclusion coming from the monotonicity of  $F$ . So  $\bigcup_{c \in V(G)} f(G_c^r) \subseteq F(G)$ . Moreover, for any  $c$ ,  $f'(G_c^r) = f'(i(G_c^r)_c^r)$ , so  $f'(G_c^r) \subseteq \bigcup_{c \in V(G)} f(G_c^r)$  and  $F(G) \subseteq \bigcup_{c \in V(G)} f(G_c^r)$ .  $\square$

**Proposition 5.** *CGD are not necessarily monotonic.*

*Proof.* Take  $|\pi| = 1$ ,  $\Sigma = \Delta = \emptyset$ . In a graph  $G$ , we have isolated vertices and pairwise connected vertices. For the local rule, consider  $r = 1$ , so the two possible disks (modulo renaming) are the isolated vertex and a pair of connected vertices. There is no graph  $G$  such that the two disks appear together (otherwise  $G$  would ask the vertex to be connected and disconnected at the same time). So we can define  $f$  such that it acts inconsistently on the two disks, since property 4 of local rules does not apply here. But the isolated vertex is included in the connected vertices in the sense of  $\subseteq$ , but the image by  $f$ , then by  $F$  are not.  $\square$

## B Proofs for Section 4 (Universality of Monotonic Causal Graph Dynamics)

**Proposition 6.** *For any graph  $G \in \mathcal{G}$ ,  $\omega_r(G_v^r) = \omega(G)_v^r$ .*

*Proof.* Take  $G_v^r = (D, v)$ . After inlining the definition of  $\omega_r$ , we are left to show  $\omega(D)_v^r = \omega(G)_v^r$ . For any entity (vertices, labels, edges) at distance at most  $r$  of  $v$ ,  $\omega(D)$  and  $\omega(G)$  coincide. At distance  $r + 1$ , things may differ for the special label  $\star$  and loopback edges, and beyond distance  $r + 1$ ,  $\omega(D)$  has no entities at all. However, all these differences are exactly what is removed by  $(-)_v^r$ . So, we get the expected equality.  $\square$

**Proposition 7.**  *$f'$  is a monotonic local rule.*

*Proof.* We first show that  $f'$  is a local rule, that is, to check the four properties of 5.

We need to show that for any isomorphism  $R$ , there is a conjugate  $R'$  such that  $f' \circ R = R' \circ f'$ . We show that the conjugate  $R'$  of  $R$  for  $f$  works. The result is obvious for  $\text{coh}(D') \in \text{Im}(\omega_{r'})$ . In the other case, take  $E' := \text{coh}(D')$ :

$$\begin{aligned}
f'(R(D'), c) &= f'((R(D'), R(c))) \\
&= \omega\left(\bigcup_{v \in R(D')_{R(c)}^{2r+1}} f(\omega_r^{-1}(R(E')_{v_i}^r))\right)_{f(\omega_r^{-1}(R(E')_{R(c)}^r))} \\
&= \omega\left(\bigcup_{v \in R(D')_{R(c)}^{2r+1}} R'(f(\omega_r^{-1}(E'_{R'(v)}^r)))\right)_{R'(f(\omega_r^{-1}(E_c^r)))} \\
&= R'\left(\omega\left(\bigcup_{v \in D_c^{2r+1}} f(\omega_r^{-1}(E_v^r))\right)\right)_{R'(f(\omega_r^{-1}(E_c^r)))} \\
&= R'\left[\omega\left(\bigcup_{v \in D_c^{2r+1}} f(\omega_r^{-1}(E_v^r))\right)_{f(\omega_r^{-1}(E_c^r))}\right] \\
&= R'(f(D', c)).
\end{aligned}$$

Consider a family of disks  $\{(D'_i, c_i)\}_{i \in I} \subset \mathcal{D}'_{r'}$  such that  $\bigcap_i D'_i = \emptyset$ . We need to show that  $\bigcap_i f'(D'_i, c_i) = \emptyset$ . We suppose that for all  $i$ ,  $E'_i := \text{coh}(D'_i) \in \text{Im}(\omega_{r'})$  (otherwise the result is trivial). First notice that for any family  $\{v_i \in (D'_i)_{c_i}^{2r+1}\}$ , we have  $\bigcap_i \omega_r^{-1}((E'_i)_{v_i}^r) = \emptyset$ , so  $\bigcap_i f(\omega_r^{-1}((E'_i)_{v_i}^r)) = \emptyset$ . Suppose now some vertex  $u$  in  $\bigcap_i f'(D'_i, c_i)$ . So, for each  $i$  there is some  $v_i$  such that  $u \in f(\omega_r^{-1}((E'_i)_{v_i}^r))$  which is impossible.

We need a bound  $b$  such that for all  $(D', c) \in \mathcal{D}'_{r'}$ ,  $|V(f'(D', c))| \leq b$ . We consider the bound  $b$  given by  $f$  and show that it works. Indeed the last step of the computation of  $f'$  is precisely a restriction to the vertices of  $f(\omega_r^{-1}(\text{coh}(D')_c^r))$ . So  $|V(f'(D', c))| \leq |V(f(\omega_r^{-1}(\text{coh}(D')_c^r)))| \leq b$ .

Consider  $G' \in \mathcal{G}'$  and  $c, d \in V(G')$ . We need  $f'(G_c^{r'})$  and  $f'(G_d^{r'})$  to be consistent. Once again, we suppose that  $\text{coh}(G_c^{r'}) \in \text{Im}(\omega_{r'})$  and  $\text{coh}(G_d^{r'}) \in \text{Im}(\omega_{r'})$  (otherwise the result is trivial). In order to use the consistency property of  $f$ , we build  $G = \omega_r^{-1}(\text{coh}(G_c^{r'})) \cup \omega_r^{-1}(\text{coh}(G_d^{r'}))$ . So for any pair of  $r$ -disks  $(D_1, D_2)$  of  $G$ ,  $f(D_1)$  and  $f(D_2)$  are consistent. In other words, all the original  $r$ -disks involved in  $f'(G_c^{r'})$  and  $f'(G_d^{r'})$  are consistent with each other. It is particularly true for  $\phi(G_c^{r'})$  and  $\phi(G_d^{r'})$ . Edges of  $f'(G_c^{r'})$  and  $f'(G_d^{r'})$  are consistent because either they come from  $f(G_c^r)$  and  $f(G_d^r)$  which are consistent, or they are backloop edges added by  $\omega$ . Labels are also consistent for the same kind of reason, making  $f'(G_c^{r'})$  and  $f'(G_d^{r'})$  consistent.

We finally show that  $f'$  is monotonic. Take two disks  $D'_1$  and  $D'_2$  such that  $D'_1 \subseteq D'_2$ . As  $f'$  is ultimately defined by cases, let us consider first the case where  $\text{coh}(D'_1) \notin \text{Im}(\omega_{r'})$ . In this case  $f'(D'_1) = \emptyset$  so we necessarily have  $f'(D'_1) \subseteq f'(D'_2)$ . We are left with the case  $\text{coh}(D'_1) \in \text{Im}(\omega_{r'})$ , meaning that  $\text{coh}(D'_1)$  is a

total disk. Since  $\text{coh}$  is monotonic by Lem. 4,  $\text{coh}(D'_1) \subseteq \text{coh}(D'_2)$ . And as a total disk, nothing but incoherences can be added to  $\text{coh}(D'_1)$ . So  $\text{coh}(D'_1) = \text{coh}(D'_2)$  and therefore  $f'(D'_1) = f'(D'_2)$ . So the order is preserved in all cases, and  $f'$  is monotonic.  $\square$