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-04511147v2

HAL Id: hal-04511147
https://hal.science/hal-04511147v2

Preprint submitted on 23 Dec 2024

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
e-mail address: luidnel.maignan@u-pec.fr, antoine.spicher@u-pec.fr

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. 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.

Introduction

Initial Motivation. This work started as an effort to understand the framework of Causal Graph Dynamics (CGDs) from the point of view of Global Transformations (GTs), both frameworks expanding on Cellular Automata (CAs) with the similar goal of handling dynamical spaces, but with two different answers. Indeed, we have on the one hand CGDs that have been introduced in 2012 in [AD12] 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 [AMP20] and quantumness [AM17]. On the other hand, we have GTs that have been proposed in 2015 in [MS15] 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

Key words and phrases: Global Transformations, Causal Graph Dynamics, Kan Extensions, Category Theory, Dynamical Systems, Model of Computations.

This publication was made possible through the support of the ID #62312 grant from the John Templeton Foundation, as part of the 'The Quantum Information Structure of Spacetime' Project (QISS) . The opinions expressed in this project/publication are those of the author(s) and do not necessarily reflect the views of the John Templeton Foundation. We also acknowledge the support of INRIA Saclay through the funding of two years of "delegation" of the first author to work on these topics.

Preprint submitted to
Logical Methods in Computer Science

© L. Maignan and A. Spicher
Creative Commons

the language of category theory. It should therefore be the case that CGDs are special cases of GTs 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 CGDs (recalled in the next section), it is appropriate to study them in the same way that CAs had been studied in [FMS23]. By this, we mean that although GTs use the language of category theory, the categorical considerations all simplify into considerations from order theory when an absolute position system is used, as is the case in CAs and in CGDs. Secondly, CGDs come directly with an order from the notion of sub-graph used implicitly throughout. The initial plan was to unfold the formalism on this basis to make sure that everything is as straightforward as expected, 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 [FMS23] for CAs.

Section 1.2 recalls how CGDs are defined and work. Section 1.3 recalls how Kan extensions and GTs are defined and simplifies in the particular case of order theory. Section 2.1 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 CGDs. The expected relationship is in fact partial and only allows to accommodate Kan extensions with CGDs which happen to be monotonic. This led us to change our initial plan to investigate the role played by monotonic CGDs within the framework. Doing so, we uncover the universality of monotonic CGDs among general CGDs, thus implying the initially wanted result: all CGDs are GTs. We then take into account renaming-invariance in a categorical way: by considering port-graphs that differ only by their names to be isomorphic, which amounts to a quotienting of the simple order-like category into a smaller category.

Sections 2.2 and 2.3 make clear the relationship between GTs and monotonic CGDs, while Section 3 shows the universality of monotonic CGDs by proposing an encoding transforming any CGD into a monotonic one. Section 4 is devoted to deal with renaming-invariance in a categorical way. Finally, Section 5 discusses these results.

1. Preliminaries

1.1. **Notations.** Sections 1, 2 and 3 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 $f \restriction A' : A' \rightharpoonup B$.

Section 4 uses elements of category theory. The reader is expected to be familiar with the notions of category, functor, natural transformation, colimit and comma category [Mac13].

1.2. **Causal Graph Dynamics.** Our work strongly relies on the objects defined in [AD13]. We recall here the definitions required to understand the rest of the paper. Particularly, [AD13] 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 1.1 (Labeled Graph with Ports)**.** Let Σ and Δ be two sets, and π a finite set. A *graph G with states in Σ and Δ, and ports in π* 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 $\sigma(G) : V(G) \rightharpoonup \Sigma$ labeling vertices of G with states,
- a partial function $\delta(G) : E(G) \rightharpoonup \Delta$ labeling edges of G with states.

The set of graphs with states in Σ and Δ, and ports π 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 $\mathrm{card}(\pi)$, the size of the finite set π. 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 Definition 1.6. This is the reason of the definition of the following object.

Definition 1.2 (Renaming)**.** A *renaming*[1] is the data of a bijection $R : \mathcal{V} \to \mathcal{V}$. 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

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

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 1.3 (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

$$V(G \cup H) := V(G) \cup V(H), \qquad E(G \cup H) := E(G) \cup E(H),$$
$$\sigma(G \cup H) := \sigma(G) \cup \sigma(H), \qquad \delta(G \cup H) := \delta(G) \cup \delta(H).$$

[1]Renamings are originally called isomorphisms. To avoid ambiguity, we reserve the use of the latter for categorical purpose.

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

$$V(G \cap H) := V(G) \cap V(H), \qquad E(G \cap H) := E(G) \cap E(H),$$
$$\sigma(G \cap H) := \sigma(G) \cap \sigma(H), \qquad \delta(G \cap H) := \delta(G) \cap \delta(H).$$

The *empty graph* with no vertex is designated $\varnothing$.

In this definition, the unions of functions seen as relations (that result is a non-functional relation in general) are here guarantied 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 1.4 (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

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

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

$$G_C^r := \bigcup_{c \in C} G_c^r. \tag{1.1}$$

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

Definition 1.5 (Local Rule)**.** A *local rule of radius r* is a function $f : \mathcal{D}^r_{\Sigma,\Delta,\pi} \to \mathcal{G}_{\Sigma,\Delta,\pi}$ such that

(1) for any renaming R, there is another renaming R', called the *conjugate* of R, with $f \circ R = R' \circ f$,
(2) for any family $\{(H_i, v_i)\}_i \subset \mathcal{D}^r_{\Sigma,\Delta,\pi}$, $\bigcap_i H_i = \varnothing \implies \bigcap_i f((H_i, v_i)) = \varnothing$,
(3) there exists b such that for all $D \in \mathcal{D}^r_{\Sigma,\Delta,\pi}$, $\mathrm{card}(V(f(D))) \leq b$,
(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 [AD13] 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 1.6 (Causal Graph Dynamics (CGD))**.** A function $F : \mathcal{G}_{\Sigma,\Delta,\pi} \to \mathcal{G}_{\Sigma,\Delta,\pi}$ is a *causal graph dynamics* (CGD) if there exists a radius r and a local rule f of radius r such

that

$$F(G) = \bigcup_{v \in V(G)} f(G_v^r). \tag{1.2}$$

An important property of CGDs is their renaming-invariance which is inherited from the properties of local rules.

Proposition 1.7 (Renaming Invariance)**.** *Let $F : \mathcal{G}_{\Sigma,\Delta,\pi} \to \mathcal{G}_{\Sigma,\Delta,\pi}$ be a CGD. For any renaming R, F admits at least a conjugate renaming R',* i.e.*, $F \circ R = R' \circ F$. Moreover, for any local rule f of F, $\mathrm{Conj}_f(R) \subseteq \mathrm{Conj}_F(R)$ where $\mathrm{Conj}_F(R)$ (resp. $\mathrm{Conj}_f(R)$) is the set of conjugates of a renaming R for F (resp. f).*

1.3. **Global Transformations & Kan Extensions.** In category theory [Mac13], Kan extensions are a construction allowing to extend a functor along another one in a universal way. In the first part 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. (The general case is given in Section 4.)

Definition 1.8 (Pointwise Left Kan Extension for Posets)**.** Given three posets A, B and C, and two monotonic functions $i : A \to B$, $f : A \to C$, the function $\Phi : B \to C$ given by

$$\Phi(b) = \sup \{\, f(a) \in C \mid a \in A \text{ s.t. } i(a) \preceq b \,\} \tag{1.3}$$

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 (GTs) make use of left Kan extensions to tackle the question of the synchronous deterministic local transformation of *arbitrary* kinds 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. (1.3), 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 Φ 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 GTs which will play an important role in the present work.

Remark 1.9. 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 Φ uses to produce output $\Phi(b')$, itself richer than output $\Phi(b)$. However, Φ cannot deduce the falsety 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. Morphisms 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 [MS15].

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

2. Unifying Causal Graph Dynamics and Kan Extensions

The starting point of our study is that Eq. (1.2) in the definition of CGDs has almost the same form as Eq. (1.3). Indeed, if we take Eq. (1.3) and set $A = \mathcal{D}^r_{\Sigma,\Delta,\pi}$, $B = C = \mathcal{G}_{\Sigma,\Delta,\pi}$, the function i to be the pointer dropping function from discs to graphs that drops their centers (*i.e.*, $i((H,c)) = H$) and the function f to be the local rule from discs to graphs, we obtain an equation for Φ of the form

$$\Phi(G) = \sup \{\, f(D) \in \mathcal{G}_{\Sigma,\Delta,\pi} \mid D \in \mathcal{D}^r_{\Sigma,\Delta,\pi} \text{ s.t. } i(D) \preceq G \,\} \tag{2.1}$$

which is close to Eq. (1.2) rewritten

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

This brings many questions. What is the partial order involved in Eq. (2.1)? Is the union of Eq. (1.2) given by the suprema of this partial order? Is it the case that $i(D) \preceq G$ implies $D = G^r_v$ for some v in this order? Are f and F of Eq. (1.2) monotonic functions for this order? We tackle these questions in the following sections.

2.1. 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 2.1 (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 2.2. *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 that 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. □

Remark 2.3. Note that this order is respected by the action of any renaming on the collection of all graphs. In other words, $R : \mathcal{G}_{\Sigma,\Delta,\pi} \to \mathcal{G}_{\Sigma,\Delta,\pi}$ is monotonic for any renaming R. The underlying reason is that it basically acts elementwise on the set $V(G)$ of each graph G, a monotonic operation for set inclusion.

The two first questions being answered positively, let us rewrite Eq. (2.1) as

$$\Phi(G) = \bigcup \{\, f(D) \mid D \in \mathcal{D}^r_{\Sigma,\Delta,\pi} \text{ s.t. } i(D) \subseteq G \,\}. \tag{2.2}$$

2.2. **Comparing Disks and Subgraphs.** Let us embark on the third question: is it the case that $i(D) \subseteq G$ implies $D = G^r_v$ 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. (1.2), the only considered disk centered on v is G^r_v. Let us determine now what are exactly the disks centered on v involved in Eq. (2.2), that is, the set $I_v := \{(H, v) \in \mathcal{D}^r_{\Sigma,\Delta,\pi} \mid H \subseteq G\}$. Firstly, G^r_v is one of them of course.

Lemma 2.4. *For any vertex $v \in V(G)$, $G^r_v \in I_v$.*

Proof. In Definition 1.4, the graph component H of the pointed graph G^r_v is explicitly defined by taking a subset for each of the four components of G, as required in Definition 2.1 of subgraph. □

The concern is that G^r_v is generally not the only one disk in I_v as expected by Eq. (1.2). However, G^r_v is the maximal one in the following sense.

Lemma 2.5. *For any vertex $v \in V(G)$, consider the disk $G^r_v = (H, v)$. Then for any disk $(H', v) \in I_v$, we have $H' \subseteq H$.*

Proof. By Definition 2.1 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. □

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 facts.

Lemma 2.6. *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 $\mathrm{card}(\pi)$ neighbors, so a rough bound is $\mathrm{card}(\pi)^r$ 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 $r \geq \mathrm{card}(V(G)) - 1$, the length of the longest possible path. □

Proposition 2.7. *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 Lemma 2.6, (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 Lemmas 2.4 and 2.5. □

We now know that the union of Eq. (2.2) receives a bigger set of local outputs to merge than the union of Eq. (1.2). 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.

2.3. **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.

2.3.1. *Monotonic CGDs 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 2.8. *Let $F : \mathcal{G}_{\Sigma,\Delta,\pi} \to \mathcal{G}_{\Sigma,\Delta,\pi}$ be a CGD with local rule $f : \mathcal{D}^r_{\Sigma,\Delta,\pi} \to \mathcal{G}_{\Sigma,\Delta,\pi}$ of radius r. If f is monotonic, then F is the pointwise left Kan extension of f along $i : \mathcal{D}^r_{\Sigma,\Delta,\pi} \to \mathcal{G}_{\Sigma,\Delta,\pi}$, the pointer dropping function.*

Proof. The proposition is equivalent to show that $F(G) = \Phi(G)$ for all G. We proceed by double-inclusion. Summarizing our journey up to here, we now know that Eq. (2.2) and Eq. (1.2) 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 Proposition 2.4. We get $F(G) \subseteq \Phi(G)$. Moreover, by Proposition 2.5, 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. □

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

Proposition 2.9. *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 Proposition 2.8 and is monotonic as recalled in Section 1.3.

Conversely, suppose F monotonic. By Definition 1.6 of CGDs, 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 $f((H,c)) = \bigcup_{v\in H} f'(H^r_v)$. We now show that f generates F. For any graph G:

$$\bigcup_{c\in V(G)} f(G^r_c) = \bigcup_{c\in V(G)} \bigcup_{v\in G^r_c} f'(i(G^r_c)^r_v).$$

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

Corollary 2.10. *A CGD is monotonic iff it is a left Kan extension.*

2.3.2. *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 CGDs, or if those CGDs 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 [AMP20].

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 on each vertex, say l and r for left and right respectively, while the particle is represented with the presence of a label on some vertex, with two possible values indicating its direction. See Figure 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 Figure 1), and wants to go to the right. If there is an outgoing edge to the right to an unlabeled vertex u (in red in Figure 1), the label representing the particle is moved from v to u (second row of Figure 1). If there is no outgoing edge to the right (the port r of v is free), it bounces by becoming a left-going particle (third row of Figure 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 Figure 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 Figure 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 Figure 1 for an illustration.

Proposition 2.11. *CGDs are not necessarily monotonic.*

Proof. Take $\mathrm{card}(\pi) = 1$, $\Sigma = \Delta = \emptyset$. 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

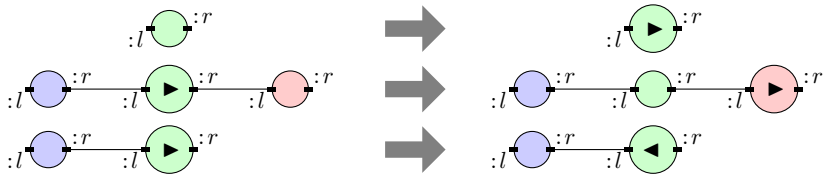

Figure 1: Moving particle CGD - non-monotonic behavior. Each row represents an example of evolution with a graph G on the left and $F(G)$ on the right. Colors correspond to vertex names.

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. The isolated vertex is included in the connected vertices in the sense of $\subseteq$, but the image by f, then by F are not. □

From Corollary 2.10, we conclude that all non-monotonic CGDs are not left Kan extensions as we have developed so far, *i.e.*, based on the subgraph relationship of Def 2.1. Analyzing the particle CGD in the light of Remark 1.9 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 Figure 1 for instance). Following Remark 1.9, 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.

3. Universality of Monotonic Causal Graph Dynamics

We have proven that the set of all CGDs 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 CGDs are universal among general CGDs.

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

3.1. **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.

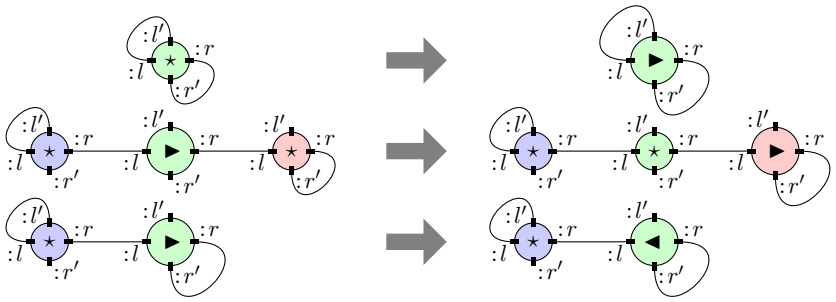

Figure 2: Moving particle CGD - monotonic behavior. Compared to Figure 1, the vertices have two additional ports (l' and r') and unlabeled vertices are now labeled by $\star$.

3.1.1. *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 l' and r' 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 $\star$. Figure 2 depicts the same evolutions as Figure 1 after those transformations.

This will be the exact role of the encoding function ω: 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.

3.1.2. *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 ω, 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 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 exist 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. Figure 3 illustrates the three identified classes: the middle graph is a coherent subgraph of the total graph on the left, and of an incoherent graph on the right. 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.

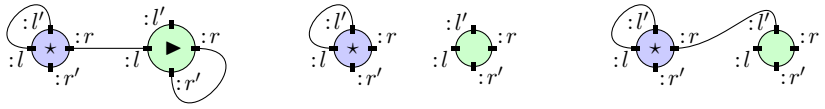

Figure 3: Different classes of graphs: from left to right, a total graph, a coherent partial graph, an incoherent graph.

3.1.3. *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 ω 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: $f'(D) = \varnothing$. (This solution corresponds to the so-called *coarse extension* proposed for CAs in [FMS23]; *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 ω. 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': $f'(D) = f'(D')$.

3.1.4. *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 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 Figure 4.

3.2. **Formal Definition of the Simulation.** Now that we have all the components of the solution, let us make them precise.

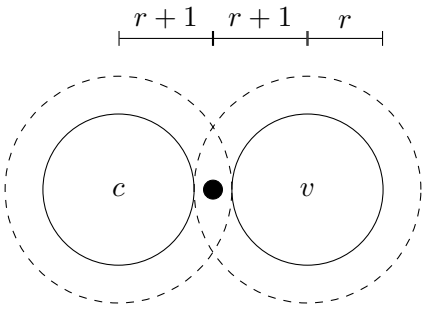

Figure 4: We need to determine for each missing entity attached to a vertex or edge of $f(G_c^r)$, whether all other $f(G_v^r)$ agree to consider this entity as missing. The property 2 of local rules (Definition 1.5) tells us that only disks G_v^r that intersect G_c^r need to be checked. The furthest 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).

3.2.1. *The Encoding Function ω.* 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 there 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 $z : X \to Z$, we define the total function $f!z : X \to Y \cup Z$ by

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

For simplicity, we write $f!\star$ for $f!(x \mapsto \star)$ 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.

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

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

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

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 $\mathrm{trg}_G^\star$.

3.2.2. *The Disk Encoding Function ω_r.* Let us now define the shorthands $\mathcal{D}_r$ and $\mathcal{D}'_r$ for $\mathcal{D}^r_{\Sigma,\Delta,\pi}$ and $\mathcal{D}^r_{\Sigma',\Delta',\pi'}$ 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 $\omega_r : \mathcal{D}_r \to \mathcal{D}'_r$ aims at encoding original disks as total disks. It is defined as

$$\omega_r((H, v)) = \omega(H)^r_v. \tag{3.1}$$

The reason we apply $(-)^r_v : \mathcal{G}' \to \mathcal{D}'_r$ on the raw result of ω is that the behavior of ω (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, ω_r is the unique function having the following commutation property, which expresses that ω_r is the disk counterpart of ω.

Proposition 3.1. *For any graph $G \in \mathcal{G}$, $\omega_r(G^r_v) = \omega(G)^r_v$.*

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

3.2.3. *Coherent Subgraph Function* 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 $\mathrm{coh} : \mathcal{G}' \to \mathcal{G}'$ that removes the incoherencies.

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

Since coh only removes incoherent edges, and since they are stable by inclusion, we trivially have that

Lemma 3.2. coh *is monotonic.*

3.2.4. *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 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 ω_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 $\phi : \mathcal{D}'_{r'} \to \mathcal{G}$.

$$\phi((H', c)) := \begin{cases} \varnothing & \text{if } \mathrm{coh}(H') \notin \mathrm{Im}(\omega_{r'}) \\ f(\omega_r^{-1}(\mathrm{coh}(H')^r_c)) & \text{otherwise} \end{cases} \tag{3.2}$$

Turning this result in $\mathcal{G}$ into its counterpart in $\mathcal{G}'$ is more complicated than simply invoking ω since one has to check that missing entities are really missing, as discussed in Figure 4.

This is the purpose of the second stage leading to the definition of the monotonic local rule $f' : \mathcal{D}'_{r'} \to \mathcal{G}'$ which, thanks to its radius $r' = 3r + 2$, is able to inspect all r-disks at distance $2r + 2$ from c.

$$f'((H', c)) = \omega \left(\bigcup_{v \in H'^{2r+1}_c} \phi((H', v)) \right)^0_{\phi((H',c))} \tag{3.3}$$

In this equation, the union of all local results that can contribute to entities attached to $\phi((H', c))$ is built. In this way, they are all given a chance to say if what was missing in $\phi((H', c))$ is actually missing. Finally ω is applied and its result restricted to the sole vertices of $\phi((H', c))$ and their edges, using the notation $(-)^0_X$ of Eq. (1.1). Those vertices and edges are the only ones for which all the possibly contributing disks have been inspected.

Proposition 3.3. *f' is a monotonic local rule.*

Proof. We first show that f' is a local rule, that is, it checks the four properties of Definition 1.5.

We need to show that for any renaming 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 $\mathrm{coh}(H') \notin \mathrm{Im}(\omega_{r'})$. In the other case, take $E' := \mathrm{coh}(H')$:

$$\begin{aligned}
f'(R((H', c))) &= f'((R(H'), R(c))) \\
&= \omega \left(\bigcup_{v \in R(H')^{2r+1}_{R(c)}} f(\omega_r^{-1}(R(E')^r_v)) \right)_{f(\omega_r^{-1}(R(E')^r_{R(c)}))} \\
&= \omega \left(\bigcup_{v \in R(H')^{2r+1}_{R(c)}} R'(f(\omega_r^{-1}(E'^r_{R'(v)}))) \right)_{R'(f(\omega_r^{-1}(E'^r_c)))} \\
&= R' \left[\omega \left(\bigcup_{v \in H'^{2r+1}_c} f(\omega_r^{-1}(E'^r_v)) \right) \right]_{R'(f(\omega_r^{-1}(E'^r_c)))} \\
&= R' \left[\omega \left(\bigcup_{v \in H'^{2r+1}_c} f(\omega_r^{-1}(E'^r_v)) \right)_{f(\omega_r^{-1}(E'^r_c))} \right] \\
&= R'(f((H', c))).
\end{aligned}$$

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

We need a bound b such that for all $(H', c) \in \mathcal{D}'_{r'}$, $\mathrm{card}(V(f'((H', c)))) \leq b$. 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 $f(\omega_r^{-1}(\mathrm{coh}(H')^r_c))$. So $\mathrm{card}(V(f'((H', c)))) \leq \mathrm{card}(V(f(\omega_r^{-1}(\mathrm{coh}(H')^r_c)))) \leq b$.

Consider $G' \in \mathcal{G}'$ and $c, d \in V(G')$. We need $f'(G'^{r'}_c)$ and $f'(G'^{r'}_d)$ to be consistent. Once again, we suppose that $\mathrm{coh}(G'^{r'}_c) \in \mathrm{Im}(\omega_{r'})$ and $\mathrm{coh}(G'^{r'}_d) \in \mathrm{Im}(\omega_{r'})$ (otherwise the result

is trivial). In order to use the consistency property of f, we build $G = \omega_r^{-1}(\mathrm{coh}(G'^{r'}_c)) \cup \omega_r^{-1}(\mathrm{coh}(G'^{r'}_d))$. 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 $f'(G'^{r'}_c)$ and $f'(G'^{r'}_d)$ are consistent with each other. It is particularly true for $\phi(G'^{r'}_c)$ and $\phi(G'^{r'}_d)$. Edges of $f'(G'^{r'}_c)$ and $f'(G'^{r'}_d)$ are consistent because either they come from $f(G^r_c)$ and $f(G^r_d)$ which are consistent, or they are loopback edges added by ω. Labels are also consistent for the same kind of reason, making $f'(G'^{r'}_c)$ and $f'(G'^{r'}_d)$ consistent.

We finally show that f' is monotonic. Take two disks $D'_1 = (H'_1, c)$ and $D'_2 = (H'_2, c)$ such that $D'_1 \subseteq D'_2$. As f' is ultimately defined by cases, let us consider first the case where $\mathrm{coh}(H'_1) \notin \mathrm{Im}(\omega_{r'})$. In this case $f'(D'_1) = \varnothing$ so we necessarily have $f'(D'_1) \subseteq f'(D'_2)$. We are left with the case $\mathrm{coh}(H'_1) \in \mathrm{Im}(\omega_{r'})$, meaning that $\mathrm{coh}(H'_1)$ is a total disk. Since coh is monotonic by Lemma 3.2, $\mathrm{coh}(H'_1) \subseteq \mathrm{coh}(H'_2)$. And as a total disk, nothing but incoherences can be added to $\mathrm{coh}(H'_1)$. So $\mathrm{coh}(H'_1) = \mathrm{coh}(H'_2)$ and therefore $f'(D'_1) = f'(D'_2)$. So the order is preserved in all cases, and f' is monotonic. □

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

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

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

Proposition 3.4. *F' is monotonic.*

Proof. Since f' is monotonic by Proposition 3.3, F' is monotonic as well by Proposition 2.9. □

Proposition 3.5. *F' simulates F via the ω 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 Proposition 3.1, the expression of $F'(\omega(G))$ simplifies drastically to

$$F'(\omega(G)) = \bigcup_{c \in V(G)} \omega\left(\bigcup\nolimits_{v \in G_c^{2r+1}} f(G^r_v)\right)^0_{f(G^r_c)}. \tag{3.4}$$

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 $f(G^r_c)$. So it belongs to the inner union of Eq. (3.4). It is preserved by ω then by $(-)^0_{f(G^r_c)}$, 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 $\ell = \delta(f(G^r_v))(u)$. So the inner union of Eq. (3.4) labels u by ℓ. The label is preserved by ω then by $(-)^0_{f(G^r_c)}$ since $u \in f(G^r_c)$. So $\delta(F'(\omega(G)))(u) = \ell$ as well. If $\ell = \star$, this means that none of the $f(G^r_v)$ put a label on u, so u is unlabeled in the inner union of Eq. (3.4). By definition, ω completes the labeling by $\star$ at u, and $(-)^0_{f(G^r_c)}$ 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. □

4. Renaming-Invariance Categorically

In Sections 2 and 3, we successfully establish the wanted connection between the fact that CGDs are particular cases of synchronous local dynamical system with absolute positioning and the fact that they can be presented as posetal GTs, *i.e.*, posetal left Kan extensions. We now want to incorporate the renaming-invariance property of CGDs. This property forbids CGDs to depend on the absolute positioning. While it is treated above as a property on top of the monotonicity of the dynamics, we now show how it can be integrated in the very algebraic structure of the graphs. This is done by turning the poset of graphs into a category of graphs where renamed graphs are considered isomorphic. This new formulation permits, in particular, to make explicit the finite nature of the local rule by expressing directly that there are finitely many disks, but that each of them can possibly be found in many places in a given graph.

Since we consider categories instead of posets, the goal is now to show that CDGs are categorical left Kan extensions. Let us recall the definition of the later and fix some notations.

Definition 4.1 (Pointwise Left Kan Extension)**.** Given three categories $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, and two functors $i : \mathbf{A} \to \mathbf{B}$, $f : \mathbf{A} \to \mathbf{C}$, a *pointwise left Kan extension of f along i* is any functor $F : \mathbf{B} \to \mathbf{C}$ such that

$$F \cong \mathrm{Colim}(f \circ \mathrm{Proj}_{i/-}) \tag{4.1}$$

when the right hand side is well defined (some colimits might not exist).

Here, $\mathrm{Colim}\, D$ designates the colimit of the functor D and is the counterpart of the supremum of Eq. (1.3). Secondly, the notation i/b stands for the comma category of the functor i over an object $b \in \mathbf{B}$ and $i/- : \mathbf{B} \to \mathbf{Cat}$ is the functor associated with this construction. It is the counterpart of the (partial ordered) set of $\{a \in A \mid i(a) \subseteq b\}$ of Eq. (1.3). However, objects (resp. morphisms) of i/b are pairs whose first components are objects (resp. morphisms) of $\mathbf{A}$, and second components is a witness of relation with b through i. The functor $\mathrm{Proj}_{i/b} : i/b \to \mathbf{A}$ thus extracts the first component of those pairs.

For the rest of the section, we fix once and for all a **monotonic** CGD F with **monotonic** local rule f of radius r. Our strategy is then as follows.

(1) Extending the poset $(\mathcal{G}_{\Sigma,\Delta,\pi}, \subseteq)$ to a category $\mathbf{G}_{\Sigma,\Delta,\pi}$ where graphs related by renaming are isomorphic. This gives more morphisms to work with on top of the original poset morphisms.
(2) Extending the monotonic function $F : \mathcal{G}_{\Sigma,\Delta,\pi} \to \mathcal{G}_{\Sigma,\Delta,\pi}$ to a functor $\widetilde{F} : \mathbf{G}_{\Sigma,\Delta,\pi} \to \mathbf{G}_{\Sigma,\Delta,\pi}$, which means specifying how the additional morphisms are dealt with through renaming-invariance.
(3) Extending the poset of disk $\mathcal{D}^r_{\Sigma,\Delta,\pi}$ to a category $\mathbf{D}^r_{\Sigma,\Delta,\pi}$ to relate disks via renamings.
(4) Extending the monotonic local rule function $f : \mathcal{D}^r_{\Sigma,\Delta,\pi} \to \mathcal{G}_{\Sigma,\Delta,\pi}$ (resp. the pointer dropping function $i : \mathcal{D}^r_{\Sigma,\Delta,\pi} \to \mathcal{G}_{\Sigma,\Delta,\pi}$) to a local rule functor $\widetilde{f} : \mathbf{D}^r_{\Sigma,\Delta,\pi} \to \mathbf{G}_{\Sigma,\Delta,\pi}$ (resp. a pointer dropping functor $\widetilde{i} : \mathbf{D}^r_{\Sigma,\Delta,\pi} \to \mathbf{G}_{\Sigma,\Delta,\pi}$).
(5) Showing that $\widetilde{F} \cong \mathrm{Colim}\, \widetilde{f} \circ \mathrm{Proj}_{\widetilde{i}/-}$.

4.1. **The Category of Graphs with Renamings.** Our first task consists in extending the previously used poset $(\mathcal{G}_{\Sigma,\Delta,\pi}, \subseteq)$ into a category allowing to compare graphs by renaming. Adding such renamings to the poset calls for a closure of the collection of *graph morphisms* (inclusions and renamings) by composition. This means that a morphism is in fact a (finite) composition of inclusions and renamings. One can convince oneself that such a composition encode the same information as a single renaming followed by an inclusion, leading to the following definition.

Definition 4.2 ((Global) Subgraph Isomorphisms)**.** Let G and H be graphs of $\mathcal{G}_{\Sigma,\Delta,\pi}$. A *(global) subgraph isomorphism* $m : G \to H$ is the data of a renaming denoted $|m| : \mathcal{V} \to \mathcal{V}$ such that $|m|(G) \subseteq H$.

Definition 4.3. We define the category $\mathbf{G}_{\Sigma,\Delta,\pi}$ whose objects are graphs of $\mathcal{G}_{\Sigma,\Delta,\pi}$ and morphisms are subgraph isomorphisms. The identity $\mathrm{id}_G : G \to G$ is given by the identity renaming (*i.e.*, $|\mathrm{id}_G| := \mathrm{id}_{\mathcal{V}}$) and the composition by composition of the underlying renamings (*i.e.*, $|n \circ m| := |n| \circ |m|$).

Proposition 4.4. $\mathbf{G}_{\Sigma,\Delta,\pi}$ *is indeed a category.*

Proof. Let $m : G \to H$ and $n : H \to K$ be two subgraph isomorphisms. We check that $|n| \circ |m|$ defines a subgraph isomorphism from G to K. We have $|m|(G) \subseteq H$ and $|n|(H) \subseteq K$, so $(|n| \circ |m|)(G) = |n|(|m|(G)) \subseteq |n|(H) \subseteq K$ as required, where we use Remark 2.3 for the first inclusion. We easily check that the identity renaming $\mathrm{id}_{\mathcal{V}}$ defines a subgraph isomorphism for any graph G ($|\mathrm{id}_G|(G) = \mathrm{id}_{\mathcal{V}}(G) = G \subseteq G$), which is neutral for the composition of renamings. Finally, associativity of the morphism composition is inherited from the associativity of renaming composition. □

In the following, we say morphism for subgraph isomorphism.

Remark 4.5. The notation $|-|$ defines in fact a faithful functor from $\mathbf{G}_{\Sigma,\Delta,\pi}$ to $\mathrm{Sym}_{\mathcal{V}}$, the symmetric group of $\mathcal{V}$ seen as a category with a single object. Interestingly, the empty graph $\varnothing$ is the unique graph accepting all renamings as endomorphisms. This provides us with a left adjoint $\varnothing_-$ to $|-|$ which maps each renaming $R \in \mathrm{Sym}_{\mathcal{V}}$ to the corresponding endomorphism $\varnothing_R : \varnothing \to \varnothing$, *i.e.*, $|\varnothing_R| = R$.

Proposition 4.6. *Isomorphisms in* $\mathbf{G}_{\Sigma,\Delta,\pi}$ *are of the form* $m : G \to |m|(G)$*, and their inverses are* $m^{-1} : |m|(G) \to G$ *such that* $|m^{-1}| = |m|^{-1}$.

Proof. We have $|m^{-1}| = |m|^{-1}$ by functoriality of $|-|$ (Remark 4.5). Suppose now that $m : G \to H$ an isomorphism and its inverse $m^{-1} : H \to G$. Let us prove that $H = |m|(G)$ by double-inclusion. For a first direction, by Definition 4.2, the morphism $m : G \to H$ tells us that $|m|(G) \subseteq H$. For the other direction, the morphism $m^{-1} : H \to G$ tells us that $|m^{-1}|(H) \subseteq G$ which rewrites into $|m|^{-1}(H) \subseteq G$ then the wanted $H \subseteq |m|(G)$ by Remark 2.3. □

In fact, each renaming R induces for each graph G an isomorphism, written $G_R : G \to R(G)$, such that $|G_R| = R$. Notice that this notation complies with the definition of functor $\varnothing_-$ of Remark 4.5. However G_- is not a functor when $G \neq \varnothing$.

We end by formally observing how $\mathbf{G}_{\Sigma,\Delta,\pi}$ embeds $(\mathcal{G}_{\Sigma,\Delta,\pi}, \subseteq)$ as we expected.

Definition 4.7 (Embedding Functor)**.** We define the *embedding functor* $\mathrm{U} : (\mathcal{G}_{\Sigma,\Delta,\pi}, \subseteq) \to \mathbf{G}_{\Sigma,\Delta,\pi}$ acting as the identity on objects and where $\mathrm{U}(G \subseteq H) : G \to H$ is such that $|\mathrm{U}(G \subseteq H)| := \mathrm{id}_{\mathcal{V}}$.

Proposition 4.8. U *is indeed an embedding functor.*

Proof. It is trivial to show that U is a functor. It is also trivially injective on objects. Finally, U is faithful since its domain is posetal. □

As expected, any morphism $m : G \to H$ can be decomposed into a renaming followed by an inclusion: $m = \mathrm{U}(|m|(G) \subseteq H) \circ (G_{|m|} : G \to |m|(G))$.

4.2. **CGDs as Endofunctors.** Now that we have extended the poset $\mathcal{G}_{\Sigma,\Delta,\pi}$ to a category $\mathbf{G}_{\Sigma,\Delta,\pi}$, our next objective is to promote the monotonic CGD F on $\mathcal{G}_{\Sigma,\Delta,\pi}$ to an endofunctor $\widetilde{F}$ on $\mathbf{G}_{\Sigma,\Delta,\pi}$. Particularly, we want $\widetilde{F}(G) = F(G)$ for all G, and $\widetilde{F}$ to reflect the monotonicity of F (*i.e.*, to map inclusions to inclusions). Those properties are captured by the following commutation.

$$\mathrm{U} \circ F = \widetilde{F} \circ \mathrm{U} \tag{4.2}$$

Let us investigate the behavior of such a $\widetilde{F}$ on morphisms $m : G \to H$. We first show that the underlying renaming of $\widetilde{F}(m)$ does not depend on G and H.

Proposition 4.9. *Consider some $\widetilde{F}$ obeying to Eq.* (4.2)*. For any $m : G \to H$, the renamings $|\widetilde{F}(m)|$ and $|\widetilde{F}(\varnothing_{|m|})|$ are equal.*

Proof. For any $m : G \to H$, since $\varnothing \subseteq G$ and $\varnothing \subseteq H$, we observe the commutation

$$m \circ \mathrm{U}(\varnothing \subseteq G) = \mathrm{U}(\varnothing \subseteq H) \circ \varnothing_{|m|} \tag{4.3}$$

where $\varnothing_{|m|} : \varnothing \to \varnothing$ is given by Remark 4.5. We get

$$\begin{aligned}
|\widetilde{F}(m)| &= |\widetilde{F}(m)| \circ \mathrm{id}_{\mathcal{V}} = |\widetilde{F}(m)| \circ |\mathrm{U}(F(\varnothing \subseteq G))| \\
&= |\widetilde{F}(m)| \circ |\widetilde{F}(\mathrm{U}(\varnothing \subseteq G))| && \text{by Eq. (4.2)} \\
&= |\widetilde{F}(m \circ \mathrm{U}(\varnothing \subseteq G))| = |\widetilde{F}(\mathrm{U}(\varnothing \subseteq H) \circ \varnothing_{|m|})| && \text{by Eq. (4.3)} \\
&= |\widetilde{F}(\mathrm{U}(\varnothing \subseteq H))| \circ |\widetilde{F}(\varnothing_{|m|})| \\
&= |\mathrm{U}(F(\varnothing \subseteq H))| \circ |\widetilde{F}(\varnothing_{|m|})| && \text{by Eq. (4.2)} \\
&= \mathrm{id}_{\mathcal{V}} \circ |\widetilde{F}(\varnothing_{|m|})| = |\widetilde{F}(\varnothing_{|m|})|
\end{aligned}$$

as required. □

This proposition tells us that the entire behavior of $\widetilde{F}$ on morphisms is simply determined by a function over renamings. We denote this function $\overline{F} : \mathrm{Sym}_{\mathcal{V}} \to \mathrm{Sym}_{\mathcal{V}}$ which is defined by $\overline{F}(R) := |\widetilde{F}(\varnothing_R)|$. Notice the commutation $\overline{F}(|m|) = |\widetilde{F}(m)|$ which holds for any $m : G \to H$ by Proposition 4.9.

Proposition 4.10. *Consider some $\widetilde{F}$ obeying to Eq.* (4.2)*. $\overline{F} : \mathrm{Sym}_{\mathcal{V}} \to \mathrm{Sym}_{\mathcal{V}}$ is a group homomorphism.*

Proof. By definition, $\overline{F}$ behaves exactly as $\widetilde{F}$ on the endomorphisms of $\varnothing$. Since $\widetilde{F}$ is functorial, $\overline{F}$ is an endofunctor on the group $\mathrm{Sym}_{\mathcal{V}}$, so a group homomorphism. □

The previous results can be summarized in the following commuting diagram.

$$\begin{array}{ccccc} \mathcal{G}_{\Sigma,\Delta,\pi} & \xrightarrow{\mathrm{U}} & \mathbf{G}_{\Sigma,\Delta,\pi} & \xrightarrow{|-|} & \mathrm{Sym}_{\mathcal{V}} \\ \downarrow{\scriptstyle F} & & \downarrow{\scriptstyle \widetilde{F}} & & \downarrow{\scriptstyle \overline{F}} \\ \mathcal{G}_{\Sigma,\Delta,\pi} & \xrightarrow{\mathrm{U}} & \mathbf{G}_{\Sigma,\Delta,\pi} & \xrightarrow{|-|} & \mathrm{Sym}_{\mathcal{V}} \end{array}$$

The following proposition states how $\overline{F}$ is related to the CGDs renaming-invariance we want to capture.

Proposition 4.11. *Consider some $\widetilde{F}$ obeying to Eq.* (4.2)*. For any renaming R, $\overline{F}(R)$ is a conjugate of R,* i.e.*, for any G, we have*

$$F(R(G)) = \overline{F}(R)(F(G)).$$

Proof. Consider the isomorphism $G_R : G \to R(G)$ with $|G_R| = R$. By functoriality of $\widetilde{F}$, $\widetilde{F}(G_R) : F(G) \to F(R(G))$ is an isomorphism as well. By Proposition 4.6, $F(R(G)) = |\widetilde{F}(G_R)|(F(G)) = \overline{F}(|G_R|)(F(G)) = \overline{F}(R)(F(G))$ as required. □

As a summary, defining $\widetilde{F}$ consists in choosing, for each renaming R, a conjugate $\overline{F}(R) \in \mathrm{Conj}_F(R)$ in a functorial way. Proposition 1.7 of CGDs renaming-invariance states that each $\mathrm{Conj}_F(R)$ is never empty. However, they might not be singletons, in which case the challenge is to make a coherent choice to obtain a functor. We put this challenge aside and place ourselves in the simpler case where all conjugates are unique (*i.e.*, $\mathrm{Conj}_F(R)$ is a singleton for any R) which we call *unique conjugate assumption*. We will discuss later the general case.

Proposition 4.12. *Under the unique conjugate assumption, Eq.* (4.2) *characterizes uniquely the functor $\widetilde{F} : \mathbf{G}_{\Sigma,\Delta,\pi} \to \mathbf{G}_{\Sigma,\Delta,\pi}$.*

Proof. By Proposition 4.9, the functor $\widetilde{F}$ is entirely defined by the behavior of $\overline{F}$. By Proposition 4.11, we have $\overline{F}(R) \in \mathrm{Conj}_F(R)$ which is a singleton under the unique conjugate assumption. So, $\widetilde{F}$ is all fixed in that case and it remains to show that it is a functor. But this means that $\overline{F} : \mathrm{Sym}_{\mathcal{V}} \to \mathrm{Sym}_{\mathcal{V}}$ is functorial, which is exactly stated by Proposition 4.10. □

4.3. **The Category of Disks.** Like it has been done for graphs, disks which are related by renaming need to be considered as isomorphic. This leads to consider a category from the poset $\mathcal{D}^r_{\Sigma,\Delta,\pi}$. This category inherits from $\mathcal{D}^r_{\Sigma,\Delta,\pi}$ the fact that disks are compared by aligning their centers. Consequently, the comma category $\widetilde{i}/G$ to be considered latter (whatever $\widetilde{i}$ will be) will not be able to identify how two disks of G with different centers can be related to each other. However, this information is of main importance when computing the colimit. Indeed, in the original poset setting, images of non aligned disks are well positioned thanks to the absolute positioning. But this is lost when quotienting by renamings. In a more general categorical setting, alignment has to be specified through morphisms directly. This is a usual issue when dealing with GTs: rules have to specify not only the images of the disks but also how these images must be connected in the output.

We need to complete the quotienting with more information about disks interactions. Past experiences in GTs show that there exist two main strategies to keep track of disks relationships: completing the category of disks either with unions of disks or intersections of disks [FMS19]. We used the former in the case of CA [FMS23] where a category of *pairs of*

disks is considered. In the present work, we focus on the latter strategy since the empty graph provides us with a common "subdisk" able to relate all proper disks.

Definition 4.13 (Category of Disks)**.** Let $\mathbf{D}^r_{\Sigma,\Delta,\pi}$ be the category with as objects the set

$$\{(H, \{c\}) \mid (H, c) \in \mathcal{D}^r_{\Sigma,\Delta,\pi}\} \cup \{(\varnothing, \emptyset)\},$$

and, for any two such objects (H_1, C_1) and (H_2, C_2), a morphism $m : (H_1, C_1) \to (H_2, C_2)$ is the data of a graph morphism (also abusively denoted) $m : H_1 \to H_2 \in \mathbf{G}_{\Sigma,\Delta,\pi}$ such that $|m|(C_1) \subseteq C_2$. Composition is inherited from $\mathbf{G}_{\Sigma,\Delta,\pi}$, and $\mathrm{id}_{(H,C)} = \mathrm{id}_H$, for any object (H, C).

This category contains as objects all disks of $\mathcal{D}^r_{\Sigma,\Delta,\pi}$ with an additional "not centered empty disk". We use a set (singleton or empty set) to deal with the presence / absence of center and get an homogeneous representation. Thanks to this encoding, a morphism from the empty disk does not require any constraint on the center ($|m|(\emptyset) = \emptyset \subseteq C_2$ whatever C_2 is), while a morphism between two proper disks requires them to be aligned by centers ($|m|(\{c_1\}) \subset \{c_2\} \iff |m|(c_1) = c_2$).

Proposition 4.14. $\mathbf{D}^r_{\Sigma,\Delta,\pi}$ *is indeed a category.*

Proof. For any $(H, C) \in \mathbf{D}^r_{\Sigma,\Delta,\pi}$, $\mathrm{id}_{(H,C)} = \mathrm{id}_H$ is well defined since we have $|\mathrm{id}_H|(C) = \mathrm{id}_{\mathcal{V}}(C) = C \subseteq C$. The composition is well defined as well. Indeed, given $m : (H, C) \to (H', C')$ and $n : (H', C') \to (H'', C'')$, we have

$$|m|(C) \subseteq C' \wedge |n|(C') \subseteq C'' \implies |n \circ m|(C) = |n|(|m|(C)) \subseteq C''$$

which implies that $n \circ m$ is a morphism of $\mathbf{D}^r_{\Sigma,\Delta,\pi}$. Associativity of the composition and neutrality of the identities are inherited from $\mathbf{G}_{\Sigma,\Delta,\pi}$ and already proved in Proposition 4.4. □

Notice that, under the common condition that the number of labels in Σ and Δ is finite, this category of disks is now essentially finite. Indeed, because of the bounds imposed by the radius and the number of ports, there is a maximal number n of vertices. If we fix a subset $W \subset \mathcal{V}$ of size n and restrict to the disks such that $V(H) \subseteq W$, we have only finitely many disks as soon as there are finite labels, and any other disk is isomorphic to one of those disks. In other words, the skeleton of the category of disks of radius r has finitely many objects. Because a category is equivalent to its skeleton, we do not make the latter explicit and continue the work with the category $\mathbf{D}^r_{\Sigma,\Delta,\pi}$.

4.4. **The Pointer Dropping and Local Rule Functors.** The next step is straightforward definitions of the categorical counterpart of functions i and f. On the one hand, the functor $\widetilde{i}$ drops the centers of disks.

Definition 4.15 (Pointer Dropping Functor)**.** Let $\widetilde{i} : \mathbf{D}^r_{\Sigma,\Delta,\pi} \to \mathbf{G}_{\Sigma,\Delta,\pi}$ be the functor defined by $\widetilde{i}((H, C)) = H$ and $\widetilde{i}(m : (H, C) \to (H', C')) = m$.

Proposition 4.16. $\widetilde{i}$ *is indeed a functor.*

Proof. The proposition is trivial since $\widetilde{i}$ simply drops centers. □

Let now exhibit the structure induced on the comma category $\widetilde{i}/G$. Formally, an object of $\widetilde{i}/G$ is a couple $\langle (H,C), m : H \to G\rangle$ composed of a disk $(H,C) \in \mathbf{D}^r_{\Sigma,\Delta,\pi}$ together with a morphism $m : H \to G$ expressing an occurrence of that disk in G. To avoid redundancies in notations, we simply write $\langle C, m : H \to G\rangle$. Morphisms of $\widetilde{i}/G$ are couples $\langle n : (H_1,C_1) \to (H_2,C_2), m : H_2 \to G\rangle : \langle C_1, m \circ n\rangle \to \langle C_2, m\rangle$. Notice that once given the specification of the couple, the signature of the morphism is completely known. In the following, signatures of morphisms are then omitted.

Proposition 4.17. *For any $G \in \mathbf{G}_{\Sigma,\Delta,\pi}$, the comma category $\widetilde{i}/G$ is non-empty and connected.*

Proof. For non-emptiness, we consider the empty disk $(\varnothing,\emptyset)$ which occurs in any graph G. So we have $\langle \emptyset, \mathrm{U}(\varnothing \subseteq G)\rangle$ is $\widetilde{i}/G$.

To show the connectiveness of $\widetilde{i}/G$, consider two arbitrary disks (H_1,C_1) and (H_2,C_2) occurring in G by $\langle C_1, m_1 : H_1 \to G\rangle$ and $\langle C_2, m_2 : H_2 \to G\rangle$. Both are related to $\langle \emptyset, \mathrm{U}(\varnothing \subseteq G)\rangle$ by the morphisms $\langle n_k : (\varnothing,\emptyset) \to (H_k,C_k), m_k\rangle$ with $|n_k| = |m_k|^{-1}$ for $k \in \{1,2\}$. Indeed, firstly n_k is well defined since $|n_k|(\emptyset) = \emptyset \subseteq C_k$ and $|n_k|(\varnothing) = \varnothing \subseteq H_k$. Secondly, $\mathrm{U}(\emptyset \subseteq G) = m_k \circ n_k$ since $|m_k \circ n_k| = |m_k| \circ |n_k| = |m_k| \circ |m_k|^{-1} = \mathrm{id}_{\mathcal{V}}$. So we have a path of length 2 in $\widetilde{i}/G$ between any two arbitrary objects $\langle C_1, m_1\rangle$ and $\langle C_2, m_2\rangle$. □

On the other hand, the local rule functor $\widetilde{f}$ applies f on disks and does nothing for the empty disk.

Definition 4.18 (Local Rule Functor)**.** Let $\widetilde{f} : \mathbf{D}^r_{\Sigma,\Delta,\pi} \to \mathbf{G}_{\Sigma,\Delta,\pi}$ be the functor defined by $\widetilde{f}((H,\{c\})) = f((H,c))$ and $\widetilde{f}((\varnothing,\emptyset)) = \varnothing$, and for any $m : (H,C) \to (H',C')$, $|\widetilde{f}(m)| = \overline{F}(|m|)$.

Proposition 4.19. *$\widetilde{f}$ is indeed a functor.*

Proof. We first need to show that $\widetilde{f}$ is well defined, *i.e.*, that $\widetilde{f}(m)$ is a morphism from $\widetilde{f}((H,C))$ to $\widetilde{f}((H',C'))$. So we expect $|\widetilde{f}(m)|(\widetilde{f}((H,C))) \subseteq \widetilde{f}((H',C'))$, which rewrites into $|\overline{F}(|m|)(\widetilde{f}((H,C))) \subseteq \widetilde{f}((H',C'))$ by definition of $\widetilde{f}$. Let us look at the three possible cases. If $(H,C) = (H',C') = (\varnothing,\emptyset)$, $\widetilde{f}((H,C)) = \widetilde{f}((H',C')) = \varnothing$ and $\overline{F}(|m|)(\varnothing) = \varnothing \subseteq \varnothing$. If $(H,C) = (\varnothing,\emptyset)$ and $(H',C') = (H',\{c'\})$, $\widetilde{f}((H,C)) = \varnothing$ and $\widetilde{f}((H',C')) = f((H',c'))$; so, $\overline{F}(|m|)(\varnothing) = \varnothing \subseteq f((H',c'))$. If $(C,H) = (H,\{c\})$ and $(H',C') = (H',\{c'\})$, $\widetilde{f}((H,C)) = f((H,c))$ and $\widetilde{f}((H',C')) = f((H',c'))$; so, we need to check that $\overline{F}(|m|)(f((H,c))) \subseteq f((H',c'))$. Observe that if $\mathrm{Conj}_f(|m|) = \{\overline{F}(|m|)\}$, we get the result: since $|m|(c) = c'$ and $|m|(H) \subseteq H'$, we have $\overline{F}(|m|)(f((H,c))) = f((|m|(H),|m|(c))) \subseteq f((H',c'))$ by monotony of f. It remains to show the assumed condition. By Definition 1.5 of local rules, f has a conjugate for $|m|$. So, by Proposition 1.7, we get $\emptyset \subsetneq \mathrm{Conj}_f(R) \subseteq \mathrm{Conj}_F(R)$ for any renaming R. By unique conjugate assumption, f has the same conjugate as F for $|m|$. So $\mathrm{Conj}_f(|m|) = \{\overline{F}(|m|)\}$.

It remains to show functoriality which is simply obtained by functoriality of $\overline{F}$ (Proposition 4.10). □

4.5. **CGDs with Unique Conjugate as Kan Extensions.** We end our journey with the main result of the section stating that $\widetilde{F}$ is the pointwise left Kan extension of $\widetilde{f}$ along $\widetilde{i}$. To get the result, we first focus on an input $G \in \mathbf{G}_{\Sigma,\Delta,\pi}$ and a colimit over $\widetilde{f} \circ \mathrm{Proj}_{\widetilde{i}/G}$, i.e. a universal cocone from $\widetilde{f} \circ \mathrm{Proj}_{\widetilde{i}/G}$ to $F(G)$.

Definition 4.20. Let $\theta^G : \widetilde{f} \circ \mathrm{Proj}_{\widetilde{i}/G} \Rightarrow F(G)$ be the cocone with components

$$\theta^G_{\langle C, m:H\to G\rangle} : \widetilde{f}((C,H)) \to F(G)$$

such that $|\theta^G_{\langle C,m:H\to G\rangle}| := \overline{F}(|m|)$.

Proposition 4.21. *θ^G is indeed a cocone.*

Proof. We need to show that the components are well-defined morphisms, then that they commute with the diagram morphisms. Consider some $\langle C, m : H \to G\rangle$ of $\widetilde{i}/G$. We want to show that $\theta^G_{\langle C,m\rangle}$ is well defined, which means that $|\theta^G_{\langle C,m\rangle}|(\widetilde{f}((C,H))) \subseteq F(G)$, that is, $\overline{F}(|m|)(\widetilde{f}((C,H))) \subseteq F(G)$ by Definition 4.20. Let us proceed by case. For $(H,C) = (\varnothing,\emptyset)$, it rewrites to $\overline{F}(|m|)(\varnothing) = \varnothing \subseteq F(G)$ which holds trivially. For $(H,C) = (H,\{c\})$, it rewrites into $\overline{F}(|m|)(f((H,c))) \subseteq F(G)$. Since $\overline{F}(|m|) \in \mathrm{Conj}_f(|m|)$, we get $\overline{F}(|m|)(f((H,c))) = f((|m|(H),|m|(c)))$. Since $m : H \to G$ and $(H,c) \in \mathcal{D}^r_{\Sigma,\Delta,\pi}$, $(|m|(H),|m|(c)) \subseteq G^r_c$. By monotonicity of f then by Eq. (1.2), $f((|m|(H),|m|(c))) \subseteq f(G^r_v) \subseteq F(G)$ as required.

For commutation, take some morphism $\langle n : (H_1,C_1) \to (H_2,C_2), m : H_2 \to G\rangle$ in $\widetilde{i}/G$. We have to check $\theta^G_{\langle C_1,m\circ n\rangle} = \theta^G_{\langle C_2,m\rangle} \circ \widetilde{f}(n)$. Since the signature is correct, it remains to check the equality of the associated renaming.

$$\begin{aligned}
|\theta^G_{\langle C_1,m\circ n\rangle}| &= \overline{F}(|m\circ n|) && \text{by Def. 4.20}\\
&= \overline{F}(|m|)\circ\overline{F}(|n|) && \text{by Prop. 4.10 and Def. 4.3}\\
&= |\theta^G_{\langle C_2,m\rangle}| \circ |\widetilde{f}(n)| && \text{by Def. 4.20 and Def. 4.18}\\
&= |\theta^G_{\langle C_2,m\rangle} \circ \widetilde{f}(n)| && \text{by Def. 4.3}
\end{aligned}$$

□

Proposition 4.22. *θ^G is a universal cocone.*

Proof. Suppose a cocone $\xi : \widetilde{f} \circ \mathrm{Proj}_{\widetilde{i}/G} \Rightarrow G'$. We have to show the existence of a unique morphism $\lambda : F(G) \to G'$ such that for all $\langle C, m : H \to G\rangle$, $\xi_{\langle C,m\rangle} = \lambda \circ \theta^G_{\langle C,m\rangle}$. If such morphism exists, it is uniquely defined by its associated renaming $|\lambda|$. Each $\langle C,m\rangle$ for $\widetilde{i}/G$ comes with the constraint that $|\lambda| = |\xi_{\langle C,m\rangle}| \circ |\theta^G_{\langle C,m\rangle}|^{-1}$. Let us set $\lambda^{\langle C,m\rangle} := |\xi_{\langle C,m\rangle}| \circ |\theta^G_{\langle C,m\rangle}|^{-1}$, and show that $\lambda^{\langle C_1,m_1\rangle} = \lambda^{\langle C_2,m_2\rangle}$ for any $\langle C_1, m_1 : H_1 \to G\rangle$ and $\langle C_2, m_2 : H_2 \to G\rangle$. To do so, since $\widetilde{i}/G$ is connected by Proposition 4.17, it is enough to notice that it holds when $\langle C_1,m_1\rangle$ and $\langle C_2,m_2\rangle$ are related by a morphism in $\widetilde{i}/G$. Suppose a morphism $\langle n : (H_1,C_1) \to (H_2,C_2), m : H_2 \to G\rangle$. We have $\lambda^{\langle C_1,m\circ n\rangle} = \lambda^{\langle C_2,m\rangle}$. Indeed, notice that since ξ and θ^G are cocones, we have $\xi_{\langle C_1,m\circ n\rangle} = \xi_{\langle C_2,m\rangle} \circ \widetilde{f}(n)$ and

$\theta^G_{\langle C_1, m\circ n\rangle} = \theta^G_{\langle C_2,m\rangle} \circ \widetilde{f}(n)$. So,

$$\begin{aligned}
\lambda^{\langle C_1, m\circ n\rangle} &= |\xi_{\langle C_1,m\circ n\rangle}| \circ |\theta^G_{\langle C_1,m\circ n\rangle}|^{-1} \\
&= |\xi_{\langle C_2,m\rangle} \circ \widetilde{f}(n)| \circ |\theta^G_{\langle C_2,m\rangle} \circ \widetilde{f}(n)|^{-1} \\
&= |\xi_{\langle C_2,m\rangle}| \circ |\widetilde{f}(n)| \circ |\widetilde{f}(n)|^{-1} \circ |\theta^G_{\langle C_2,m\rangle}|^{-1} \\
&= |\xi_{\langle C_2,m\rangle}| \circ |\theta^G_{\langle C_2,m\rangle}|^{-1} \\
&= \lambda^{\langle C_2,m\rangle}
\end{aligned}$$

as expected. Since $\widetilde{i}/G$ is not empty by Proposition 4.17 and all $\lambda^{\langle C,m\rangle}$ agree, we are able to consider λ such that $|\lambda| = \lambda^{\langle C,m\rangle}$. It remains to show that λ defined with such an associated renaming is a morphism from $F(G)$ to G'. Equivalently, we have to show that $|\lambda|(F(G)) \subseteq G'$.

$$\begin{aligned}
|\lambda|(F(G)) &= |\lambda|(\bigcup_{c\in V(G)} f(G^r_c)) \\
&= \bigcup_{c\in V(G)} |\lambda|(f(G^r_c)) \\
&= \bigcup_{c\in V(G)} \lambda^{\langle \{c\}, \mathrm{U}(i(G^r_c)\subseteq G)\rangle}(f(G^r_c)) \\
&= \bigcup_{c\in V(G)} |\xi_{\langle \{c\}, \mathrm{U}(i(G^r_c)\subseteq G)\rangle}|(|\theta^G_{\langle \{c\}, \mathrm{U}(i(G^r_v)\subseteq G)\rangle}|^{-1}(f(G^r_c))) \\
&= \bigcup_{c\in V(G)} |\xi_{\langle \{c\}, \mathrm{U}(i(G^r_c)\subseteq G)\rangle}|(\overline{F}(\mathrm{id}_{\mathcal{V}})^{-1}(f(G^r_c))) \\
&= \bigcup_{c\in V(G)} |\xi_{\langle \{c\}, \mathrm{U}(i(G^r_c)\subseteq G)\rangle}|(f(G^r_c)) \\
&\subseteq \bigcup_{c\in V(G)} G' \\
&= G'
\end{aligned}$$

The inclusion comes from the fact that $\xi_{\langle \{c\}, \mathrm{U}(i(G^r_c)\subseteq G)\rangle}$ is a morphism from $f(G^r_c)$ to G'. This concludes the proof of universality of θ^G. □

Proposition 4.23. *$\widetilde{F}$ is the pointwise left Kan extension of $\widetilde{f}$ along $\widetilde{i}$.*

Proof. For any graph G, since θ^G is a universal cocone with apex $F(G)$ by Proposition 4.22, we already have that $\widetilde{F}(G) \cong \mathrm{Colim}\, \widetilde{f} \circ \mathrm{Proj}_{\widetilde{i}/G}$. It remains to show the functoriality of the construction which means that for any $h : G \to G'$, $\widetilde{F}(h) \cong \mathrm{Colim}\, \widetilde{f} \circ \mathrm{Proj}_{\widetilde{i}/h}$. On the right side, the expressed morphism is the unique mediating λ from θ^G to $\theta^{G'}_{\widetilde{i}/h}$. To clarify, $\widetilde{i}/h$ is the functor embedding $\widetilde{i}/G$ into $\widetilde{i}/G'$ mapping any $\langle n : (H_1, C_1) \to (H_2, C_2), m : H_2 \to G\rangle$ to $\langle n, h\circ m\rangle$. So, $\theta^{G'}_{\widetilde{i}/h}$ is the cocone over $\widetilde{f}\circ\mathrm{Proj}_{\widetilde{i}/G}$ with components $\theta^{G'}_{(\widetilde{i}/h)(\langle C,m:H\to G\rangle)} = \theta^{G'}_{\langle C,h\circ m\rangle}$. By universality of θ^G (Proposition 4.22), we get a unique morphism $\lambda : F(G) \to F(G')$. Using the elements of the proof of Proposition 4.22, we use the object $\langle \emptyset, \mathrm{U}(\varnothing \subseteq G)\rangle \in \widetilde{i}/G$

to get

$$\begin{aligned}
|\lambda| &= \lambda^{\langle\emptyset,\mathrm{U}(\varnothing\subseteq G)\rangle} \\
&= |\theta^{G'}_{(\langle\emptyset,h\circ\mathrm{U}(\varnothing\subseteq G)\rangle)}| \circ |\theta^{G}_{(\langle\emptyset,\mathrm{U}(\varnothing\subseteq G)\rangle)}|^{-1} \\
&= \overline{F}(|h\circ\mathrm{U}(\varnothing\subseteq G)\rangle)|) \circ \overline{F}(|\mathrm{U}(\varnothing\subseteq G)\rangle)|)^{-1} \\
&= \overline{F}(|h|) \circ \overline{F}(|\mathrm{U}(\varnothing\subseteq G)\rangle)|) \circ \overline{F}(|\mathrm{U}(\varnothing\subseteq G)\rangle)|)^{-1} \\
&= \overline{F}(|h|) \\
&= |\widetilde{F}(h)|.
\end{aligned}$$

We conclude that $\lambda = \widetilde{F}(h)$ (both have same signature and underlying renaming) which proves the required functoriality. □

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 CGDs (for a particular partial order on graphs) which are both CGDs and GTs, and happen to be universal among all CGDs. We have then transformed the renaming-invariance property of the dynamics into additional relations between the graphs, so that the resulting notion of monotonicity (functoriality in fact) now ingrates renaming-invariance. There are at least two benefits for that. Firstly this is more inline with usual categories for graphs where no notion of absolute names or positioning exists, but only their (possibly multiple) occurrences in each other. Secondly, this new formulation permits, in the particular case where there is a finite number of labels, to make explicit the finite nature of the local rule by expressing directly that there are finitely many disks, but that each of them can possibly be found at many places in a given graph.

This is still more work to do to finish this line of reasoning though. Firstly, while the (skeleton of the) category of disks has finitely many objects, there are still infinitely many morphisms between them. This is because the chosen notion of morphisms is based on bijections of the (uncountable) set $\mathcal{V}$, in order to stay as close as possible to the framework of CGDs. It should be possible to get rid of $\mathcal{V}$ completely and have simple injective functions between arbitrary sets of vertices.

Secondly, we have only treated the case of CGDs with "unique conjugates". However, this property does not a priori holds for all CGDs. Indeed, one can show that if $R' \in \mathrm{Conj}_F(R)$, we have $\mathrm{Conj}_F(R) = R' \circ \mathrm{Conj}_F(\mathrm{id}_{\mathcal{V}})$. Now any $S \in \mathrm{Conj}_F(\mathrm{id}_{\mathcal{V}})$ is, by definition, such that for all G, $S(F(G)) = F(G)$. This means that they are symmetries, and the number of conjugates of dynamics depends on whether they admit symmetries that are common to all the outputs. A first consequence is that if any G has an output without symmetry, then F respects the unique conjugate assumption. For instance, it is the case for the moving particle CGD example used in Section 3 (see some asymmetric outputs in Figure 2). A second possible consequence might be that "unique conjugates" dynamics are somehow universal. Indeed, start with an arbitrary dynamics, the conjugates of the identity invite to make the quotient the outputs of F with respect to those symmetries, hopefully yielding a "unique conjugates" dynamics.

Coming back to Section 2. 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 CGDs 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 CGDs 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 CGDs 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 $G = H^0_{V(G)}$. 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 CGDs 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.

References

[AD12] Pablo Arrighi and Gilles Dowek. Causal graph dynamics. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 54–66. Springer, 2012. doi:10.1007/978-3-642-31585-5_9.

[AD13] Pablo Arrighi and Gilles Dowek. Causal graph dynamics. *Inf. Comput.*, 223:78–93, 2013. URL: https://doi.org/10.1016/j.ic.2012.10.019, doi:10.1016/J.IC.2012.10.019.

[AM17] Pablo Arrighi and Simon Martiel. Quantum causal graph dynamics. *Physical Review D*, 96(2):024026, 2017.

[AMP20] Pablo Arrighi, Simon Martiel, and Simon Perdrix. Reversible causal graph dynamics: invertibility, block representation, vertex-preservation. *Nat. Comput.*, 19(1):157–178, 2020. doi:10.1007/S11047-019-09768-0.

[FMS19] Alexandre Fernandez, Luidnel Maignan, and Antoine Spicher. Lindenmayer systems and global transformations. In Ian McQuillan and Shinnosuke Seki, editors, *Unconventional Computation and Natural Computation - 18th International Conference, UCNC 2019, Tokyo, Japan, June 3-7, 2019, Proceedings*, volume 11493 of *Lecture Notes in Computer Science*, pages 65–78. Springer, 2019. doi:10.1007/978-3-030-19311-9_7.

[FMS23] Alexandre Fernandez, Luidnel Maignan, and Antoine Spicher. Cellular automata and kan extensions. *Nat. Comput.*, 22(3):493–507, 2023. doi:10.1007/S11047-022-09931-0.

[Mac13] S. MacLane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer New York, 2013.

[MS15] Luidnel Maignan and Antoine Spicher. Global graph transformations. In Detlef Plump, editor, *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, volume 1403 of *CEUR Workshop Proceedings*, pages 34–49. CEUR-WS.org, 2015. URL: http://ceur-ws.org/Vol-1403/paper4.pdf.

This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, 171 Second St, Suite 300, San Francisco, CA 94105, USA, or Eisenacher Strasse 2, 10777 Berlin, Germany