Abstract

In this paper, we formalize precisely the sense in which the application of a cellular automaton to partial configurations is a natural extension of its local transition function through the categorical notion of Kan extension. In fact, the two possible ways to do such an extension and the ingredients involved in their definition are related through Kan extensions in many ways. These relations provide additional links between computer science and category theory, and also give a new point of view on the famous Curtis–Hedlund theorem of cellular automata from the extended topological point of view provided by category theory. These links also allow to relatively easily generalize concepts pioneered by cellular automata to arbitrary kinds of possibly evolving spaces. No prior knowledge of category theory is assumed for the most part.

Keywords

Cellular automata · Kan extensions · Global transformations · Curtis–Hedlund theorem

1 Introduction

Unconventional computing models take their inspirations from natural physical and living systems. One can mention Lindenmayer systems (Rozenberg and Salomaa 2012), the chemical metaphor with the CHAM formalism (Berry and Boudol 1992) and the Γ language (Banâtre et al. 2000), the biological cell metaphor with Păun systems (Păun 2001), multi-agent systems, and so on. All these models share the common property that they allow the specification of systems in a discrete and local way by the mean of rules describing how neighbor entities composing the system interact leading to an original paradigm of programming (Spicher and Giavitto 2017). However they differ in how the rules are applied (from sequentially to altogether in parallel) and in the spatial relationship structuring the system (from a lose space of a chemical system where all elements are neighbors to an arbitrary space of a multi-agent system). We particularly focus on the case where the local rules are applied synchronously and deterministically. The present work is part of a larger study seeking to investigate the relationships between local and global dynamics in such systems. We revisit the particular case of cellular automata for which these relations are already known, in view of a generalization to less regular cases, that is discussed in the conclusion.

Cellular automata are usually presented either as a local behavior extended to a global and uniform behavior, or as a continuous uniform global behavior for the appropriate topology (Ceccherini-Silberstein and Coornaert 2010; Hedlund 1969). The present work offers a third point of view tailored for generalizations of the concepts pioneered by cellular automata, via the so-called global transformations (Fernandez et al. 2019, 2021; Maignan et al. 2015).

The goal of this paper is not to elaborate on these generalizations but to focus on some simple foundational bridges allowing these generalizations, illustrated on the seminal case of cellular automata. In particular, we focus on Kan extensions, a categorical notion allowing, as we show here, to capture local/global descriptions (MacLane 2013). The goal of this paper is not to elaborate on these generalizations but to focus on some simple foundational bridges allowing these generalizations, illustrated on the seminal case of cellular automata. In particular, we focus on Kan extensions, a categorical notion allowing, as we show here, to capture local/global descriptions (MacLane 2013).
and to introduce Kan extensions in a very simple and not
categorical way. In a second time, the setting is enriched to
take into account the shift operations allowing to go from
an absolute positioning to a relative positioning. It requires
a transition to the more general setting of categories which
appears to be closer to the case of global transformations as
discussed in the conclusion.

The rest of the article is organized as follows. In Sect. 2,
we recall the direct definitions of cellular automata on
groups, local transition function, global transition function,
shift action, and also consider the counterparts of these
functions on the poset of arbitrary partial configurations.
This bigger picture allows to show that the various
local/global relations between these objects are all captured
This choice of definition and right notation for the so
called shift action has two advantages. Firstly, the defini-
tion we refer explicitly to

\[ \text{Definition 2} \] A cellular automaton on a group \( G \) is given
by a finite subset \( N \subset G \) called the neighborhood, a finite
set of states \( Q \), and a local transition function \( \delta : Q^N \to Q \).
The elements of the set \( Q^N \) are called local configurations.
The elements of the set \( Q^G \) are called global configurations
and a right action \( \cdot : Q^G \times G \to Q^G \) is defined on \( Q^G \)
by \( (c \cdot g)(h) = c(g \cdot h) \) for all \( h \in G \). The global transition
function \( \Delta : Q^G \to Q^G \) of such a cellular automaton is defined as
\( \Delta(c)(g) = \delta((c \cdot g)|N) \).

Notice that in this definition we refer explicitly to
global and local configurations; the term configuration with no
qualification will be introduced in Definition 6.

\[ \text{Proposition 3} \] The function \( \cdot : Q^G \times G \to Q^G \) of
Definition 2 is indeed a right action.

\[ \text{Proof} \] For any \( g,h \in G \), we have
\( ((c \cdot g) \cdot h)(i) = (c \cdot g)(h \cdot i) = c(g \cdot h) \cdot i = (c \cdot (g \cdot h))(i) \)
for any \( i \in G \), so
\( (c \cdot g) \cdot h = c \cdot (g \cdot h) \). Also \( (c \cdot 1)(i) = c(1 \cdot i) = c(i) \)
as required by Definition 1 of right actions.

This choice of definition and right notation for the so
called shift action has two advantages. Firstly, the defini-
tion of the action is a simple associativity. Secondly, when
instantiated with \( G = \mathbb{Z} \) with sum, the content of \( c \cdot 5 \) is the
content of \( c \) shifted to the left, as the symbols indicates.
Indeed, for \( c' = c \cdot 5 \), \( c'(-5) = c(0) \) and \( c'(0) = c(5) \).

\[ \text{Proposition 4} \] For all \( c \in Q^G \) and \( g \in G \), \( \Delta(c)(g) \) is
determined by \( c \cdot g \cdot N \).

\[ \text{Proof} \] Indeed, \( \Delta(c)(g) = \delta((c \cdot g)|N) \) so the value is
determined by \( (c \cdot g)|N \). But for any \( n \in N \), \( (c \cdot g)(n) = c(g \cdot n) \)
by definition of \( \cdot \).

In common cellular automata terms, this proposition
means that the neighborhood of \( g \) is \( g \cdot N \), in this order.
Let us informally call shifted local configurations the objects of
the form \( c \cdot g \cdot N \in \bigcup_{g \in G} Q^G \). Note that, at our level of
generality, two different positions \( g \neq g' \in G \) might have
the same neighborhood \( g : N = g' : N \).
Although the injectivity of the function \( \cdot : N \) could be a useful constraint
so the reader should keep this in mind.

\[ \text{Proposition 5} \] The function \( \cdot : N : G \to 2^G \) is not neces-
sarily injective.

\[ \text{Proof} \] Considering the group \( G = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z} \) and
\( N = \{ (0,0), (1,0) \} \), we have
\( (0,0) + N = (1,0) + N = \{ (0,0), (1,0) \} \).

Because of this, it is useful to replace the shifted local
configurations, i.e. the union \( \bigcup_{g \in G} Q^G \), by the disjoint
union \( \bigcup_{g \in G} \{ g \} \times Q^G \). The elements of the latter are of

2 Cellular automata and partial
configurations

Let us give some basic definitions to fix the notations. We
also note small caveats early on, to avoid having to deal
with many unrelated details at the same time in a single
proof or construction later on.

2.1 Cellular automata

\[ \text{Definition 1} \] A group is a set \( G \) with a binary operation
\( \cdot : G \times G \to G \) which is associative, which has a
neutral element \( 1 \) and for which any \( g \in G \) has inverse \( g^{-1} \).
A right action of the group on a set \( X \) is a binary operation
\( \cdot : X \times G \to X \) such that \( x \cdot 1 = x \) and
\( (x \cdot g) \cdot h = x \cdot (g \cdot h) \).

In cellular automata, the group \( G \) represents the space,
each element \( g \in G \) being at the same time an absolute and
a relative position. This space is decorated with states that
evolve through local interactions only. The classical formal
definitions go as follows and work with the entire, often
infinite, space.
the form \( (g \in G, c \in Q^eN) \) and keep track of the considered “center” of the neighborhood. More explicitly, two elements \((g_0, c|g_0 \cdot N), (g_1, c|g_1 \cdot N) \in \bigcup_{e \in G} \{g\} \times Q^eN\) are different as soon as \(g_0 \neq g_1\) even if \(g_0 \cdot N = g_1 \cdot N\). This encodes things according to the intuition of a centered neighborhood.

### 2.2 The poset of configurations

In the previous formal statements, one sees different kinds of configurations, explicitly or implicitly: global configurations \(c \in Q^G\), local configurations \((c|g) \n N \in Q^N\), shifted local configurations \((c \cdot g \cdot N) \in Q^N\), and their resulting “placed states” \((g \mapsto \Delta(c)(g)) \in Q^S\). In the cellular automata literature, one often considers configurations defined on other subsets of the space, e.g., finite connected subsets. More generally, we are interested in all configurations \(Q^G\) for arbitrary subsets \(S \subseteq G\). The restriction operation \([- |-]\) used many times above gives a partial ordering of these configurations.

**Definition 6** A configuration \(c\) is a partial function from \(G\) to \(Q\). Its domain of definition is denoted \(|c|\) and called its support. The set of all configurations is denoted \(\text{Conf} = \bigcup_{S \subseteq G} Q^S\). We extend the previous right action \(\triangleleft\) and define it to map each \(c \in \text{Conf}\) to \((c|g)\) having support \(|(c|g)\| = \{h \in G \mid g \cdot h \in |c|\}\) and states \((c|g)(h) = c(g \cdot h)\).

**Proposition 7** The latter right action is well-defined and is a right action.

**Proof** The configuration \((c|g)\) is well-defined on all of its support. Indeed for any \(h \in |(c|g)|\), \((c|g)(h) = c(g \cdot h)\) and \(g \cdot h \in |c|\) by definition of \(h\). The right action property is verified as in the proof of Proposition 3.

Let us restate Proposition 4 more precisely using Definition 6.

**Proposition 8** For \(c \in \text{Conf}\) and \(g \in G\) s.t. \(g \cdot N \subseteq |c|\), \((c|g) \n N = (c|g \cdot N) \triangleleft g\).

**Proof** Indeed, \(|(c|g \cdot N) \triangleleft g| = \{h \in G \mid g \cdot h \in |(c|g \cdot N)|\} = \{h \in G \mid g \cdot h \in g \cdot N\} = N = |(c|g)|\n N\). Also for any \(n \in N\), \((c|g)(N)(n) = (c|g)(n) = c(g \cdot n)\) and \((c|g \cdot N)(\triangleleft g)(n) = (c|g \cdot N)(g \cdot n) = c(g \cdot n)\).

**Definition 9** A partial order on a set \(X\) is a binary relation \(\preceq\subseteq X \times X\) which is reflexive, transitive, and antisymmetric. A set endowed with a partial order is called a partially ordered set, or poset for short.

**Definition 10** Given any two configurations \(c, c' \in \text{Conf}\), we set \(c \preceq c'\) if and only if \(\forall g \in |c|, g \in |c'| \land c(g) = c'(g)\). This is read “\(c\) is a subconfiguration of \(c'\)” or “\(c'\) is a superconfiguration of \(c\).”

**Proposition 11** The set \(\text{Conf}\) with this binary relation is a poset. In this poset, the shifted local configurations \(c \in \bigcup_{e \in G} Q^eN\) are subconfigurations of the (appropriate) global configurations \(c' \in Q^G\). Shifted local configurations form an antichain. Global configurations form an antichain.

**Proof** As can be readily seen, since each global configuration restricts to many shifted local configurations, and recalling that an antichain is a subset \(S\) of the poset such that neither \(x \preceq x'\) nor \(x' \preceq x\) hold for any two different \(x, x' \in S\).

### 3 Poset Kan extensions in cellular automata

Since posets are a particular kind of categories, Kan extensions apply naturally to the poset of configurations. So we begin by giving the definition of left and right Kan extensions in the particular case of posets. Then we generalize the global transition function into two monotonic transition functions acting on all the poset of configurations, each being then characterized as Kan extensions.

#### 3.1 Kan extensions for posets

Given three sets \(A, B, C\) such that \(A \subseteq B\), we say that a function \(g : B \rightarrow C\) extends a function \(f : A \rightarrow C\) if \(g|A = f\), or equivalently if \(g \circ i = f\) where \(i\) is the obvious injective function from \(A\) to \(B\). For a given \(f : A \rightarrow C\), there are typically many possible extensions. Roughly speaking, Kan extensions formalizes, among many things, the mathematical practice where extensions are rarely arbitrary. One usually chooses the “best” or “most natural” extensions. There is therefore an implicit comparison considered between the extensions.

This is the reason why Kan extensions are formally defined at the level of 2-categories: \(A, B, C\) are objects, \(f, g, i\), and all (not necessarily “most natural”) extensions are 1-arrows between these objects, and the “naturality” comparison between 1-arrows are 2-arrows. However, there is no need to discuss things at this level of generality for the moment. In the case of posets, the objects are posets, the 1-arrows are monotonic functions and the monotonic functions are compared pointwise.

**Definition 12** Given two posets \((X, \preceq_X)\) and \((Y, \preceq_Y)\), a function \(f : X \rightarrow Y\) is said to be monotonic if for all \(x, x' \in X\), \(x \preceq_X x'\) implies \(f(x) \preceq_Y f(x')\).
Proposition 13 For any $g \in G$, the function $(- \triangleright g) : \text{Conf} \rightarrow \text{Conf}$ is monotonic.

Proof Given any $c, c' \in \text{Conf}$ such that $c \preceq c'$, this claim is equivalent to:
\[
(c \triangleright g) \preceq (c' \triangleright g) \quad \text{(by Def 12)}
\]
\[
\Leftrightarrow \forall h \in [c \triangleright g] ; h \in [c' \triangleright g] \land (c \triangleright g)(h) = (c' \triangleright g)(h) \quad \text{(Def 10)}
\]
\[
\Leftrightarrow \forall h \in G \text{ s.t. } g \cdot h \in [c] ; g \cdot h \in [c'] \land c(g \cdot h) = c'(g \cdot h) \quad \text{(Def 6)}
\]
which is true by the application of Definition 10 of $c \preceq c'$ on $g \cdot h$.

Definition 14 Given two posets $(X, \preceq_X)$ and $(Y, \preceq_Y)$, we define the binary relation $- \Rightarrow -$ on the set of all monotonic functions from $X$ to $Y$ by $f \Rightarrow f' \iff \forall x \in X, f(x) \preceq_Y f'(x)$.

Proposition 15 Given two posets $(X, \preceq_X)$ and $(Y, \preceq_Y)$, the set of monotonic functions between them together with this binary relation forms a poset.

Proof As one can easily check.

Definition 16 In this setting, given three posets $A, B$ and $C$, and three monotonic functions $i : A \rightarrow B, f : A \rightarrow C$ and $g : B \rightarrow C$, $g$ is said to be the left (resp. right) Kan extension of $f$ along $i$ if $g$ is the $\Rightarrow$-minimum (resp. $\Rightarrow$-maximum) element in the set of monotonic functions $\{ h : B \rightarrow C \mid f \Rightarrow h \circ i \}$ (resp. $\{ h : B \rightarrow C \mid h \circ i \Rightarrow f \}$).

This concept is particularly useful because, it also has a complete characterization as stated in the following proposition.

Proposition 17 The left (resp. right) Kan extension $g$ is unique when it exists.

Proof It is defined as the minimum of a set, and as any minimum, it may not exist, but when it does, it is always unique.

Another suggestive way to read the concept of Kan extensions with respect to this paper is to say that a function $g$ on a poset can be summarized into, or generated by, a part of its behavior $f$ on just a small part of the poset. Note however that $i$ does not need to be injective in this definition.

3.2 The coarse monotonic transition function

A first way to extend the global transition function to the whole poset of partial configurations is as follows. Taking a configuration $c$, look for all places $g$ where the whole neighborhood $g \cdot N$ is defined and to take the local transition result of these places only. This shows in particular that the global transition function is the left Kan extension of the “fully shifted” local transition. The sense of “fully shifted” is described below and is only necessary because we restrict ourselves to posets, as discussed later.

3.2.1 Direct definition

Definition 18 The interior of a subset $S \subseteq G$ is $\text{int}(S) = \{ g \in G \mid g \cdot N \subseteq S \}$.

Definition 19 The coarse transition function $\Phi : \text{Conf} \rightarrow \text{Conf}$ is defined for all $c \in \text{Conf}$ as $|\Phi(c)| = \text{int}(|c|)$ and $\Phi(c)(g) = \delta((c \triangleright g) \cdot N)$.

Proposition 20 For any $c \in \text{Conf}$ and $g \in G$, the statements $g \in \text{int}(|c|)$, $g \cdot N \subseteq |c|$, and $N \subseteq |c \triangleright g|$ are equivalent. (So $\Phi$ is well-defined in Definition 19.)

Proof The first and second statements are equivalent by Definition 18 of int. The second and third statements are equivalent by Definition 6 of $\triangleright$.

Remember Proposition 5. If we do have injectivity of neighborhoods, we have $\text{int}(g \cdot N) = \{g\}$. But since we do not assume it, we only have the following.

Proposition 21 Let $S \subseteq G$. It is always the case that $S \subseteq \text{int}(S \cdot N)$ but we do not necessarily have equality, even when $S = \{g\}$ for some $g \in G$.

Proof Consider any $s \in S$. Clearly, $s \cdot N \subseteq S \cdot N$, so by Definition 18, $s \in \text{int}(S \cdot N)$. However, we do not have equality in the example of the proof of Proposition 5 with $S = \{0, 0\}$. Indeed, in this case, $\text{int}(S \cdot N) = \{(0, 0), (1, 0)\}$.

Of course, the coarse transition function shares with the global transition function the fact that it commutes with the shift.

Proposition 22 For any $c \in \text{Conf}$ and $g \in G$ we have $\Phi(c \triangleright g) = \Phi(c) \triangleright g$.

Proof We must check that $|\Phi(c \triangleright g)| = |\Phi(c) \triangleright g|$ and that for any $h \in |\Phi(c \triangleright g)|$ we have $\Phi(c \triangleright g)(h) = (\Phi(c) \triangleright g)(h)$. For the first part we have:
\[
\begin{align*}
\Phi(c \triangleright g)(h) &= \delta((c \triangleright g)(h) \cdot N) \quad \text{(Def. 19 of $\Phi$)} \\
&= \delta((c \triangleright g \cdot h) \cdot N) \quad \text{(Prop. 7)} \\
&= (\Phi(c)(g \cdot h)) \quad \text{(Def. 19 of $\Phi$)} \\
&= (\Phi(c) \triangleright g)(h) \quad \text{(Def. 6)}
\end{align*}
\]

The last proposition is true by the Proposition 7. For the second part we have:
\[
\begin{align*}
\Phi(c \triangleright g)(h) &= \delta((c \triangleright g)(h) \cdot N) \quad \text{(Def. 19 of $\Phi$)} \\
&= \delta((c \triangleright g \cdot h) \cdot N) \quad \text{(Prop. 7)} \\
&= (\Phi(c)(g \cdot h)) \quad \text{(Def. 19 of $\Phi$)} \\
&= (\Phi(c) \triangleright g)(h) \quad \text{(Def. 6)}
\end{align*}
\]
Another useful remark on which we come back below is the following.

**Proposition 23** For any \( g \in G \), any \( M \subseteq N \), and any \( c \in Q^{M} \), \( |\Phi(c)| = \emptyset \). Also, for any \( c \in Q^{G} \), \( |\Phi(c)| = |A(c)| \).

**Proof** By Definition 19 of \( \Phi \).

### 3.2.2 Characterization as a left Kan extension

The coarse transition function \( \Phi \) is defined on the set of all configurations \( \text{Conf} \) and we claim that it is generated, in the Kan extension sense, by the local transition function \( \delta \) shifted everywhere. We define the latter, with Proposition 5 in mind.

**Definition 24** We define \( \text{Loc} \) to be the poset \( \text{Loc} = \bigcup_{g \in G} \{g\} \times Q^{N} \) with trivial partial order \( (g, c) \preceq (g', c') \iff (g, c) = (g', c') \). The “fully shifted local transition function” \( \delta : \text{Loc} \to \text{Conf} \) is defined, for any \( (g, c) \in \text{Loc} \) as \( \delta(g, c) = \{g\} \) and \( \delta(g, c)(g) = \delta(c \triangleright g) \). The second projection of \( \text{Loc} \) is the monotonic function \( \pi_{2} : \text{Loc} \to \text{Conf} \) defined as \( \pi_{2}(g, c) = c \).

**Proposition 25** \( \text{Loc} \) is a poset and \( \delta \) and \( \pi_{2} \) are monotonic functions.

**Proof** Indeed, the identity relation is an order relation and any function respects the identity relation.

**Proposition 26** The coarse transition function \( \Phi \) is monotonic.

**Proof** Indeed, take \( c, c' \in \text{Conf} \) such that \( c \preceq c' \). We want to prove that \( \Phi(c) \preceq \Phi(c') \) and this is equivalent to:

\[
\forall g \in \Phi(c), g \in \Phi(c') \iff \Phi(g)(c) = \Phi(g')(c) \]

\[
\iff \forall g \in \text{int}(\{c\}), g \in \text{int}(\{c'\}) \land \delta(c \triangleright g|N) = \delta(c' \triangleright g|N) \]

\[
\iff \forall g \in G \text{ s.t. } g \cdot N \subseteq |c|, g \cdot N \subseteq |c'| \land \delta(c \triangleright g|N) = \delta(c' \triangleright g|N),
\]

by Definition 6 of \( \preceq \), Definition 19 of \( \Phi \), and Definition 18 of \( \text{int} \). The final statement is implied by:

\[
\forall g \in G \text{ s.t. } g \cdot N \subseteq |c|, g \cdot N \subseteq |c'| \land \Phi(g)(c) = \Phi(g)(c') \]

\[
\iff \forall g \in G \text{ s.t. } g \cdot N \subseteq |c|, g \cdot N \subseteq |c'| \land \forall h \in N, (c \triangleright g)(n) = (c' \triangleright g)(n) \]

\[
\iff \forall g \in G \text{ s.t. } g \cdot N \subseteq |c|, g \cdot N \subseteq |c'| \land \forall h \in N, c(g \cdot n) = c'(g \cdot n),
\]

the last equivalence being by Definition 6. To prove it, take \( g \in G \) such that \( g \cdot N \subseteq |c| \), and \( n \in N \). By Definition 10, since \( c \preceq c' \) and \( g \cdot n \in |c'| \), we have \( g \cdot n \in |c'| \), and \( c(g \cdot n) = c'(g \cdot n) \) as wanted.

**Proposition 27** \( \Phi \) is the left Kan extension of \( \delta \) along \( \pi_{2} : \text{Loc} \to \text{Conf} \).

**Proof** By Definition 16 of left Kan extensions, we need to prove firstly that \( \Phi \) is such that \( \delta \Rightarrow \Phi \circ \pi_{2} \), and secondly that it is smaller than any other such monotonic function.

For the first part, \( \delta \Rightarrow \Phi \circ \pi_{2} \) is equivalent to:

\[
\forall (g, c) \in \text{Loc}, \delta(g, c) \leq \Phi(c) \left( \text{Defs. 14 and 24 of } \Rightarrow \text{ and } \pi_{2} \right)
\]

\[
\iff \forall (g, c) \in \text{Loc}, \forall h \in |\Phi(c)| \land \delta(g, c)(h) = \Phi(c)(h) \left( \text{Def. 10 of } \leq \right)
\]

\[
\iff \forall (g, c) \in \text{Loc}, g \in \Phi(c), \delta(g, c)(g) = \Phi(g)(c) \left( \text{Def. 24 of } \delta \right)
\]

\[
\iff \forall (g, c) \in \text{Loc}, g \cdot N \subseteq |c| \land \delta(c \triangleright g) = \delta((c \triangleright g)|N) \left( \text{Defs 18, 19 of } \Phi \right).
\]

This last statement is true by Definition 24 of \( \text{Loc} \), i.e. since \( c \in Q^{N} \), \( c \triangleright g = (c \triangleright g)|N \).

For the second part, let \( F : \text{Conf} \to \text{Conf} \) be a monotonic function such that \( \delta \Rightarrow F \circ \pi_{2} \). We want to show that \( \Phi \Rightarrow F \), which is equivalent to:

\[
\forall c \in \text{Conf}, \delta(c) \leq F(c) \left( \text{Def. 14 of } \Rightarrow \right)
\]

\[
\iff \forall c \in \text{Conf}, \forall g \in \Phi(c), g \in |F(c)| \land \delta(g)(c) = F(g)(c) \left( \text{Def. 10 of } \leq \right)
\]

\[
\iff \forall c \in \text{Conf}, \forall g \in \text{int}(\{c\}), g \in |F(c)| \land \delta(g)(c) = F(g)(c)(|N) \left( \text{Def. 19} \right)
\]

So take \( c \in \text{Conf} \) and \( g \in \text{int}(\{c\}) \), and consider \( d_{g} = c[g \cdot N] \). Since \( d_{g} \preceq c \) and \( F \) is monotonic, we have \( F(d_{g}) \preceq F(c) \). Moreover \( \delta \Rightarrow F \circ \pi_{2} \) and \( (g, d_{g}) \in \{g\} \times Q^{N} \subseteq \text{Loc} \). By Definition 24 of \( \delta \) and Definition 10 of \( \leq \), we obtain \( g \in |F(c)| \), and \( F(c)(g) = \delta(g)(d_{g})(c) = \delta((c \triangleright g)|N) \), as wanted.

As a sidenote, remark that in order to have the equality \( \delta = \Phi \circ \pi_{2} \), one needs to have the injectivity of neighborhood function. Indeed, without injectivity, we have two different \( g, g' \in G \) having the same neighborhood \( M \), i.e. \( M = g \cdot N = g' \cdot N \). This means that, given any local configuration \( c \in Q^{M} \) on this neighborhood, each pair \( (g, c), (g', c') \in \text{Loc} \) have different results \( \delta(g, c) \in Q^{|x|} \) and \( \delta(g', c) \in Q^{|x'|} \) with different support \( \{g\} \) and \( \{g'\} \). However, their common projection \( \pi_{2}(g, c) = \pi_{2}(g', c') = c \) have a unique result \( \Phi(c) \) with a support such that \( \{g, g'\} \subseteq |\Phi(c)| \). So we have a strict comparison \( \delta \Rightarrow \Phi \circ \pi_{2} \). When the neighborhood function is injective, \( \pi_{2} \) is also injective and the previous situation can not occur so we have equality \( \delta = \Phi \circ \pi_{2} \).

### 3.3 The fine monotonic transition function

For some applications, the previous definitions are not satisfactory. For example, it is common to consider two cellular automata to be essentially the same if they generate the same global transition functions. However, here, two
such cellular automata give different coarse transition functions if they have a different neighborhood.

To refine the previous definitions, a second approach is to take a configuration $c$, and look at all places for which the result is already determined by the partial data defined in $c$. So we consider all $g \in G$ for which all completions of the data present on the defined neighborhood $g \cdot N \cap |c|$ into a configuration on the complete neighborhood $g \cdot N$ always lead to the same result by $\delta$.

### 3.3.1 Direct definition

Given a configuration $c$ and a position $g$, we define now $c_g$ the part of $c$ that is in the neighborhood $g$. We can consider $c_g$ as a partial function from $g \cdot N$ to $Q$. If this partial function is enough for the result at $g$ to be already determined, i.e. if any extension of $c_g$ into a function $c'$ from $g \cdot N$ to $Q$ gives the same result $\delta(c' \upharpoonright g)$, we say that $g$ is in the determined subset of $c$ and call $q_{c,g}$ the determined result.

**Definition 28** For any $c \in \text{Conf}$ and $g \in G$, let $c_g = c|\{g \cdot N \cap |c|\}$.

**Definition 29** Given a configuration $c \in \text{Conf}$, its determined subset is $\text{det}(c) = \{ g \in G | \exists q \in Q, \forall c' \in Q^N, c'||c_g = c_g \Rightarrow \delta(c' \upharpoonright g) = q \}$. For any $g \in \text{det}(c)$, we denote $q_{c,g} \in Q$ the unique state $q$ having the mentioned property.

Note that this definition depends on the cellular automaton local transition function $\delta$ and on the data of the configuration $c$, contrary to Definition 18 of interior that only depends on its neighborhood $N$ and on the support of the configuration.

**Definition 30** Given a cellular automaton, its fine transition function $\phi : \text{Conf} \rightarrow \text{Conf}$ is defined as $|\phi(c)| = \text{det}(c)$ and $\phi(c)(g) = q_{c,g}$, i.e. $\phi(c)(g) = \delta(c' \upharpoonright g)$ for any $c' \in Q^N$ such that $c'||c_g = c_g$.

**Proposition 31** The fine transition function $\phi$ is well defined.

**Proof** This is the case precisely because we restrict the support of $\phi(c)$ to the determined subset of the $c$.

**Proposition 32** Consider the constant cellular automaton $\delta(c) = q \forall c \in Q^N$ for a specific $q \in Q$ regardless of the neighborhood $N$ chosen to represent it. We have $\| \phi(c) \| = G$ for any $c \in \text{Conf}$.

**Proof** Indeed, even with no data at all, i.e. for $c$ such that $|c| = \emptyset$, the result at all position is determined and is $q$.

Note that, contrary to Proposition 23 of the coarse transition function, the fine transition function definition is explicitly about considering non-empty results for some configurations of support $M \subseteq N$ (see Definition 28). When there is no such proper “sub-local” configuration with determined result, the two transition functions (Definitions 19 and 30) are actually equal since $\text{int}(c) = \text{det}(c)$ for any $c \in \text{Conf}$ (Definitions 29 and 18). But let us not insist on this point.

### 3.3.2 Characterization as a right Kan extension

As for the coarse transition function, the fine transition function $\phi$ is defined on the set of all configurations Conf and we claim that it is generated, in the Kan extension sense. We consider two ways to generate it and start by the simplest one. The second one is considered in the following section using sub-local configurations in order to be closer to the direct definition and to be a “from local to global” characterization.

**Proposition 33** For any $g \in G$, the function $-_g : \text{Conf} \rightarrow \text{Conf}$ of Definition 28 is monotonic.

**Proof** As one can easily check.

**Proposition 34** The fine transition function $\phi$ is monotonic.

**Proof** Indeed, take $c_0, c_1 \in \text{Conf}$ such that $c_0 \preceq c_1$. We want to prove that $\phi(c_0) \preceq \phi(c_1)$ and this is equivalent to:

$$\forall g \in |\phi(c_0)|, g \in |\phi(c_1)| \land \phi(c_0)(g) = \phi(c_1)(g) \text{ (Def 10 of } \preceq)$$

$$\iff \forall g \in \text{det}(c_0), g \in \text{det}(c_1) \land q_{c_0,g} = q_{c_1,g} \text{ (Def 30 of } \phi)$$

Take $g \in \text{det}(c_0)$. We want to prove that $g \in \text{det}(c_1)$, which means by Definition 29 of $\text{det}(c_1)$:

$$\exists q \in Q, \forall c_2 \in Q^N, c_2 |||\langle c_1 \rangle_g| = \langle c_1 \rangle_g \Rightarrow \delta(c_2 \upharpoonright g) = q$$

We claim that the property is verified with $q = q_{c_0,g}$. Indeed, take any $c_2 \in Q^N$ such that $c_2 |||\langle c_1 \rangle_g| = \langle c_1 \rangle_g$. We also have that $c_2 |||\langle c_0 \rangle_g| = \langle c_0 \rangle_g$ since the hypothesis $c_0 \preceq c_1$ implies $\langle c_0 \rangle_g \preceq \langle c_1 \rangle_g$. By Proposition 33. By Definition 29 of $\text{det}(c_0)$, we obtain that $\delta(c_2 \upharpoonright g) = q_{c_0,g}$, so $q = q_{c_0,g}$ has the wanted property, which implies that $g \in \text{det}(c_1)$ as wanted. But the above property of $q$ set it to be precisely what we denote by $q_{c_1,g}$ (Definition 29 of $q_{c_1,g}$), so $q_{c_0,g} = q_{c_1,g}$.

**Proposition 35** The fine transition function $\phi$ is the right Kan extension of the global transition function $\Delta$ along the inclusion $i : Q^G \rightarrow \text{Conf}$.
Proof. By Definition 16 of right Kan extensions, we need to prove firstly that φ is such that φ ∘ i ⇒ Δ, and secondly that it is greater than any other such monotonic functions.

For the first part, we actually have φ ∘ i ⇒ Δ since for any c ∈ Qg, |φ(c)| = det(c) = |Δ(c)| and for any g ∈ G, we have φ(c)(g) = qc,g = δ(c | g) = δ(c | g · N | g) = δ((c | g) · N | g) = Δ(c)(g) by Definitions 30, 29, 28, 2 of φ, det, qc and Δ and Prop. 8.

For the second part, let f : Conf → Conf be a monotonic function such that f ∘ i ⇒ Δ. We want to show that f ⇒ φ, which is equivalent to:

∀c ∈ Conf. f(c) ≤ φ(c) (Def. 14 of ⇒)

⇐⇒ ∀c ∈ Conf. ∀g ∈ f(c), g ∈ φ(c) ∧ f(c) = φ(c)(g) (Def. 10 of ≤)

⇐⇒ ∀c ∈ Conf. ∀g ∈ f(c), g ∈ det(c) ∧ f(c) = qc,g (Def. 30 of φ)

⇐⇒ ∀c ∈ Conf. ∀g ∈ f(c), ∀c′ ∈ Qg, c ≤ c′ ⇒ φ(c)(g) = δ(c′ | g).

So take c ∈ Conf and g ∈ f(c) and c′ ∈ Qg such that c ≤ c′. Consider any c″ ∈ Qg such that c′ ≤ c″ (or equivalently c″(g · N) = c′). Since f is monotonic, we have f(c) ≤ f(c″), which means that f(c)(g) = f(c″)(g) by Definition 10. But since f ∘ i ⇒ Δ, we have f(c)(g) = Δ(c″)(g) = δ(c″ | g)|N by Definition 14 of ⇒ and Definition 2 of Δ. But by Prop. 8, (c″ | g)|N = (c″ · g) · g = c″ | g.

Proposition 36. Let us consider another cellular automaton having neighborhood N′ ⊆ G and local transition function δ' : QN′ → Q. Consider its corresponding global transition function δ' : QN → Q and fine transition function φ' : Conf → Conf. Then if δ' = Δ, then φ' = φ.

Proof. By Propositions 35 and 17, φ is determined by Δ, and φ′ by δ′. So Δ′ = Δ gives φ = φ′.

3.4 Introducing sub-local configurations

The direct definition of the fine transition function is explicitly about assigning a result for a configuration c at a given g ∈ G even when the whole neighborhood g · N is not complete. By isolating these “shifted sub-local configurations” in the poset of configurations, we can (right-)extend the local transition to them and show that, in the same way as the coarse transition function is the left Kan extension of the local transition function, the fine transition function is the left Kan extension of the sub-local transition function.

3.4.1 Direct definition

Definition 37. We define Sub = ∪g∈G M⊂N {|g| × Qg} with partial order defined as (g, c) ≤ (g′, c′) if and only if g = g′ and c ≤ c′. The “fully shifted sub-local transition function” δ : Sub → Conf is defined, for any g ∈ G, any M ⊆ N and any c ∈ Qg such that δ(c, g) = |g| ∩ det(c) and, if g ∈ det(c), δ(g, c)(g) = qg,c,g. i.e., δ(g, c)(g) = δ(c | g) for any c′ ∈ Qg such that c = c′ | g. The second projection of Sub is the function π2 : Sub → Conf defined as π2(g, c) = c.

In this definition, a given sub-local configuration can result either in an empty configuration when the transition is not determined, or in a configuration with only singleton support when the transition is determined.

Note that for a given cellular automaton, it is possible to restrict the set Sub to an antichain. Indeed, any time a result is determined by a sub-local configuration (g, c), all bigger sub-local configuration (g, c′) with c ≤ c′ does not contribute anything new. We do not elaborate on this because this antichain would be different for each cellular automaton, blurring the global picture presented below.

3.4.2 Characterization as a right Kan extension

Proposition 38. The fully shifted sub-local transition function δ is monotonic.

Proof. As usual, take (g, c), (g′, c′) ∈ Sub such that (g, c) ≤ (g′, c′). First note that g = g′ by Definition 37. We want to prove that δ(g, c) ≤ δ(g, c′) and this is equivalent to:

∀h ∈ δ(g, c), h ∈ δ(g′, c′) ∧ δ(g, c)(h) = δ(g′, c)(h)

⇐⇒ ∀h ∈ {g} ∩ det(c), h ∈ {g} ∩ det(c′) ∧ qg,c,h = qg′,c,h

⇐⇒ g ∈ det(c) ⇒ g ∈ det(c′) ∧ qg,c,h = qg′,c,h,

by Definition 6 of ≤ and Definition 37 of δ. The end of this proof is similar to the one of Proposition 34.

Proposition 39. The fully shifted sub-local transition function δ is the right Kan extension of the fully shifted local transition function δ along the inclusion i : Loc → Sub.

Proof. By Definition 16 of right Kan extensions, we need to prove firstly that δ is such that δ ∘ i ⇒ δ, and secondly that it is greater than any other such monotonic functions.

For the first part, δ ∘ i ⇒ δ is equivalent to:

∀(g, c) ∈ Loc, δ(g, c) ≤ δ(g, c) (Def. 14 of ⇒)

⇐⇒ ∀(g, c) ∈ Loc, ∀h ∈ δ(g, c), h ∈ δ(g, c) ∧ δ(g, c)(h) = δ(g, c)(h) (Def. 10 of ≤)

⇐⇒ ∀(g, c) ∈ Loc, g ∈ det(c) ⇒ g ∈ δ(g, c) ∧ qg,c,h = qg,c,h (Def. 37 of δ)

⇐⇒ ∀(g, c) ∈ Loc, g ∈ det(c) ⇒ g ∈ |g| ∧ qg,c,h = δ(c | g) (Def. 24 of δ).

This last statement is true by Definition 29 of qg,c.

For the second part, let f : Sub → Conf be a monotonic function such that f ∘ i ⇒ δ. We want to show that f ⇒ δ, which is equivalent to:

δ i
By Definition 16 of left Kan extensions, we need to find a tonic function such that
\( f(g,c) \leq f(g,c') \) (Def. 14 of \( \Rightarrow \)).

\[ \iff \forall (g,c) \in \text{Sub} \, \text{f}, \exists h \in \text{Sub} \, \text{f}(g,c) \land h \in [g,c] \land f(h) = \delta(g,c)(h) \quad (\text{Def. 10}) \]

\[ \iff \forall (g,c) \in \text{Sub} \, \text{f}, \exists h \in \text{Sub} \, \text{f}(g,c) \land h \in [g,c] \land \delta(g,c)(h) = q_{c,h} \quad (\text{Def. 37}) \]

So take \((g,c) \in \text{Sub} \land h \in \text{Sub} \, \text{f}(g,c)\). Consider any \( c' \leq c \). Since \( f \) is monotonous, we have \( f(g,c) \leq f(g,c') \), which means that \( h \in \text{Sub} \, \text{f}(g,c') \) and \( f(g,c') = f(g,c')(h) \) by Definition 10. But since \( f \circ i \Rightarrow \delta \), we have \( h \in [\delta(g,c')] = \{g\} \) and \( f(g,c')(h) = \delta(g,c')(h) = \delta(c' \rangle g) \) by Definition 14 of \( \Rightarrow \) and Definition 24 of \( \delta \). Since this is true for any \( c' \), this establishes the defining property of \( \text{det}(c) \) by Definition 29.

3.4.3 The fine transition function as a left Kan extension

**Proposition 40** The projection function \( \pi_2 : \text{Sub} \rightarrow \text{Conf} \) is monotonous.

**Proof** As can be readily checked in Definition 37.

**Proposition 41** \( \phi \) is the left Kan extension of \( \delta \) along \( \pi_2 : \text{Sub} \rightarrow \text{Conf} \).

**Proof** By Definition 16 of left Kan extensions, we need to prove firstly that \( \phi \) is such that \( \delta \Rightarrow \phi \circ \pi_2 \), and secondly that it is smaller than any other such monotonous functions.

For the first part, \( \delta \Rightarrow \phi \circ \pi_2 \) is equivalent to:

\[ \forall (g,c) \in \text{Sub} \, \delta(g,c) \leq \phi(c) \quad (\text{Defs. 14 and 37 of } \Rightarrow \text{ and } \pi_2) \]

\[ \iff \forall (g,c) \in \text{Sub} \, \exists h \in [\delta(g,c)] \land h \in \delta(c) \land \delta(c)(h) = \phi(c)(h) \quad (\text{Def. 10 of } \leq) \]

\[ \iff \forall (g,c) \in \text{Sub} \, \exists g \in \text{det}(c) = \exists h \in \text{det}(c) \land q_{c,h} = \delta(g,c)(h) \quad (\text{Def. 37 of } \delta) \]

\[ \iff \forall (g,c) \in \text{Sub} \, g \in \text{det}(c) \Rightarrow g \in \text{det}(c) \land q_{c,h} = \delta(g,c)(h) \quad (\text{Det. 30 of } \phi) \]

a most trivial statement.

For the second part, let \( f : \text{Conf} \rightarrow \text{Conf} \) be a monotonous function such that \( \delta \Rightarrow f \circ \pi_2 \). We want to show that \( \phi \Rightarrow f \), which is equivalent to:

\[ \forall c \in \text{Conf}, \phi(c) \leq f(c) \quad (\text{Def. 14 of } \Rightarrow) \]

\[ \iff \forall c \in \text{Conf}, \exists g \in \delta(c) \land g \in f(c) \land \phi(c)(g) = f(c)(g) \quad (\text{Def. 10 of } \leq) \]

\[ \iff \forall c \in \text{Conf}, \exists g \in \delta(c) \land g \in f(c) \land q_{c,h} = f(c)(g) \quad (\text{Def. 30 of } \phi) \]

So take \( c \in \text{Conf} \) and \( g \in \text{det}(c) \). Since \( c \leq c \) (Definition 28) and \( f \) is monotonic, we have \( f(c_g) \leq f(c) \). Moreover \( \delta \Rightarrow f \circ \pi_2 \) and \( (g,c_g) \in \text{Sub} \) so \( \delta(g,c_g) \leq f(c_g) \) by Definitions 14 and 24 of \( \Rightarrow \) and \( \pi_2 \). By transitivity \( \delta(g,c_g) \leq f(c) \). By Definition 37 of \( \delta \) and Definition 10 of \( \leq \), we therefore have \( g \in f(c) \), and \( f(c)(g) = \delta(g,d_g)(g) = q_{c,g} \) as wanted.

4.1 The configuration of categories

A category is a generalization of posets and pre-ordered sets (poset without the antisymmetry property) where two elements can be related to each other in different ways. The order relation \( X \times X \rightarrow 2 \) becomes a function \( X \times X \rightarrow \text{Set} \), where \( X \) stands for the set of elements. Transitivity becomes a composition operation, and reflexivity becomes an identity operation.

**Definition 42** A category\(^1\) \( C \) is composed of a set of objects \( \text{Ob} \) and a set of arrows \( \text{Hom} \) for each pair of objects \( x, y \) in \( \text{Ob} \) together with a composition operation \( \circ : \text{Hom}(y,z) \times \text{Hom}(x,y) \rightarrow \text{Hom}(x,z) \) for any \( x,y,z \in \text{Ob} \) and an identity arrow \( id_x \in \text{Hom}(x,x) \) for each \( x \in \text{Ob} \). We write \( f : x \rightarrow y \) as a notation for \( f \in \text{Hom}(x,y) \).

A pre-ordered set is precisely a category where there is either zero or one arrow between any two objects (short proof in Sect. 1.2 of MacLane 2013). In our case, from the set of configurations seen as a category with an arrow from \( c \) to \( c' \) if and only if \( c \leq c' \), we now consider additional arrows when \( c \) is found at other positions in \( c' \), i.e., an arrow for each \( g \in G \) such that \( c \leq c' \).\( g \).

**Definition 43** Let \( \text{Conf} \) be the category whose objects are the configurations of \( \text{Conf} \), and for any configurations \( c \) and \( c' \), the set of arrows \( \text{Hom}_{\text{Conf}}(c,c') \) is \( \{g \in G \mid c \leq (c' \rangle g)\} \). Arrows are thus identified with elements of the group.

Given two arrows \( g : c \rightarrow c' \) and \( g' : c' \rightarrow c'' \), their composite \( g' \circ g \) is given by \( g' \circ g \).

For each object \( c \), the identity arrow \( id_c : c \rightarrow c \) is given by the neutral element \( 1 \) of the group.

**Proposition 44** This is a well defined category.

---

\(^1\) We chose the collection of objects and arrows to be sets. They are classes in usual definitions.

\(^2\) Note that one can only compose pairs of arrows where the first arrow starts where the second arrow ends, for instance \( g \circ f \) is defined for \( g : y \rightarrow z \) and \( f : x \rightarrow y \). This means that associativity, left neutrality and right neutrality only holds when they are defined.
Proof By Definition 43, given two arrows \( g : c \to c' \) and \( g' : c' \to c'' \), we have \( c \leq (c' \bowtie g) \) and \( c' \leq (c'' \bowtie g') \). Then \((c' \bowtie g) \leq ((c'' \bowtie g') \bowtie g)\) (Prop. 13) and finally \(c \leq ((c'' \bowtie g') \bowtie g)\) which rewrites to \(c \leq (c'' \bowtie (g' \cdot g))\). This proves that the group element \(g' \cdot g\) is an arrow from \(c\) to \(c''\). Finally, the composition is associative by associativity of the group operation and the identity arrow \(\text{id}_c : c \to c\). is the neutral arrow for the composition as it is the neutral element for the group operation. \(\square\)

In order to avoid unnecessary technical details, we do not consider the behavior of the functor on configurations having no results. We use the notion of a full subcategory \(C\) of a category \(D\), i.e. a category with less objects, i.e. \(\text{Ob}_C \subseteq \text{Ob}_D\), but with all arrows between selected objects, i.e. \(\text{Hom}_C(x, y) = \text{Hom}_D(x, y)\) for all \(x, y \in C\).

Definition 45 Let \(\text{Conf}^*\) be the full subcategory of \(\text{Conf}\) that contains only super local configurations, i.e. 
\[
\text{Ob}_{\text{Conf}^*} = \{c \in \text{Conf} \mid \exists g \in G \text{ s.t. } g \cdot N \subseteq |c|\}
\]
the set of configurations that contain at least one local configuration.

Proposition 46 This is a well defined category

Proof As a full subcategory \(\text{Conf}^*\) have for objects a subset of the objects of \(\text{Conf}\) and for any two objects \(c, d \in \text{Ob}_{\text{Conf}^*}\) the set \(\text{Hom}_{\text{Conf}^*}(c, d)\) is equal to \(\text{Hom}_{\text{Conf}}(c, d)\), the composition is well defined and respects associativity and identity laws. \(\square\)

Notice that given any configuration \(c\) and any \(g \in G\), we can consider the configuration \(c \bowtie g\). Interestingly, the simple relations \(c \leq (c \bowtie g) \bowtie g^{-1}\) and \((c \bowtie g) \leq c \bowtie g\) respectively imply the existence of arrows \(g^{-1} : c \to (c \bowtie g)\) and \(g : (c \bowtie g) \to c\) by Definition 43. These arrows are inverse arrows in the sense that we have \(g^{-1} \circ g = \text{id}_{c \bowtie g}\) and \(g \circ g^{-1} = \text{id}_c\). This particular relation between \(c\) and \(c \bowtie g\) is called isomorphism, and \(c\) and \(c \bowtie g\) are said to be isomorphic. Isomorphism generalizes the notion of equivalence in a pre-order and bijections between sets. Indeed, in any category, two isomorphic objects cannot be distinguished by any categorical property.

Another remark is that the poset \(\text{Conf}\) is found in the category \(\text{Conf}^*\) as expected, in the form of arrows \(1 : c \to c\) since \(c \leq c \bowtie 1 = c\). Moreover, such an arrow \(1 : c \to c\) has no inverse arrow from \(c'\) to \(c\) in general. In fact, as a group element \(1 \in G\) is its own inverse, but as an arrow from \(c\) to \(c'\), its inverse exists only if \(c = c'\). So the reader should be careful to check whether a group element is manipulated as an arrow or not, and when it is an arrow, the reader should keep in mind the associated pair of configurations.

To work with the category \(\text{Conf}\), we need to lift monotonic functions, comparisons of monotonic functions and Kan extensions given in Sect. 3 into the categorical setting. First, monotonic functions becomes functors. While a monotonic function \(m\) sends each \(x \leq y\) to \(m(x) \leq m(y)\), a functor \(F\) needs to send each possible arrow \(f : x \to y\) to a specific arrow \(F(f) : F(x) \to F(y)\) since there are many possible choices. Moreover, \(F\) needs to respect compositions and identities.

Definition 47 Given two categories \(C\) and \(D\), a functor \(F : C \to D\) is a mapping that sends each object \(x \in C\) to an object \(F(x) \in D\) and each arrow \(f : x \to y \in C\) to an arrow \(F(f) : F(x) \to F(y)\) such that \(F(g \circ f) = F(g) \circ F(f)\) for any two arrows \(f, g \in C\), and \(F(\text{id}_x) = \text{id}_{F(x)}\) for any object \(x \in C\).

Comparison of two monotonic functions \(m\) and \(n\) simply asks to check if it is true or false that, for each \(x\), that \(m(x) \leq n(x)\) (Definition 14). So the collection of monotonic functions is a poset (Proposition 15). Lifting this idea for comparing two given functors \(F\) and \(G\) consists in choosing a specific \(\eta_x : F(x) \to G(x)\) for each object \(x\) of the category. Moreover, these choices need to be coherent with image arrows of \(F\) and \(G\). Such a specific choice of an arrow \(\eta_x\) for each \(x\) is called a natural transformation. There may exist zero, one or more ways to define natural transformations between two functors \(F\) and \(G\). In fact, the collection of functors turns to be a category with natural transformations as arrows (proof in Sect. II.4 of MacLane 2013).

Definition 48 Given two functors \(F, G : C \to D\), a natural transformation \(\eta : F \Rightarrow G\) is given by a component arrow \(\eta_x : F(x) \to G(x)\) in \(D\) for each object \(x\) in \(C\) such that \(\eta_x \circ F(f) = G(f) \circ \eta_y\) for each \(f : x \to y\) in \(C\). This last condition is called the naturality condition.

Last but not least, the definition of left and right Kan extensions in the categorical setting follows the same generalization strategy by unfolding the fact that they are minimum and maximum in the poset setting. An additional uniqueness requirement is also necessary for coherence. The composition of functors used in the following definition is the obvious one.

Definition 49 Given three categories \(A, B\) and \(C\), three functors \(I : A \to B, F : A \to C\) and \(G : B \to C\), and a natural transformation \(\eta : I \Rightarrow G \circ I\) (resp. \(\eta : G \circ I \Rightarrow F\)), the pair \((G, \eta)\) is the left (resp. right) Kan extension of \(F\) along \(I\) if for any other pair \((H : B \to C, \rho : F \Rightarrow H \circ I)\) (resp. \(\rho : H \circ I \Rightarrow F\)), there is an unique natural transformation \(\lambda : G \to H\) (resp. \(\lambda : H \to G\)) such that we have \(\lambda_{I(a)} \circ \eta_a = \rho_a\) (resp. \(\eta_a \circ \lambda_{I(a)} = \rho_a\)) for any \(a \in A\).
4.2 The coarse transition functor

The coarse monotonic transition function $\Phi$ given in Section 3.2 is defined on the poset $\text{Conf}$. We lift its definition to categories $\text{Conf} / \text{Conf}^*$ and get the coarse transition functor. In the same manner, the coarse transition functor can also be characterized as a left Kan extension.

4.2.1 Direct definition

**Definition 50** The coarse transition functor $\tilde{\Phi} : \text{Conf}^* \to \text{Conf}$ is defined for any $c \in \text{Conf}^*$ as $\Phi(c) = \Phi\circ g$ (given in Definition 19) and for any arrow $g : c \to c'$ as $\Phi(g) = g : \Phi(c) \to \Phi(c')$.

**Proposition 51** The coarse transition functor $\tilde{\Phi}$ is well defined.

**Proof** First, any object $c$ in $\text{Conf}^*$ is sent to an object $\tilde{\Phi}(c) \in \text{Conf}$.

We now check that, for any arrow $g : c \to c'$ in $\text{Conf}^*$, $\tilde{\Phi}(g)$ is a valid arrow from $\tilde{\Phi}(c)$ to $\tilde{\Phi}(c')$. By Definition 43 of arrows of $\text{Conf}$, this amounts to prove that for any $g \in G$, $c \leq (c' \bowtie g)$ implies $\Phi(c) \leq (\Phi(c') \bowtie g)$, that we obtain as follows.

$c \leq (c' \bowtie g) \Rightarrow \Phi(c) \leq \Phi(c') \bowtie g$ (by Prop 26 of monotonicity of $\Phi$)

$\Rightarrow \Phi(c) \leq (\Phi(c') \bowtie g)$ (by Prop 22)

$\Rightarrow \Phi(c) \leq \Phi(c') \bowtie g$ (by Def 50 of $\Phi$)

Finally, we have $\tilde{\Phi}(g \circ h) = g \circ h = \tilde{\Phi}(g) \circ \tilde{\Phi}(h)$ and $\tilde{\Phi}(id_c) = id_c = 1 = id_{\tilde{\Phi}(c)}$, which concludes the proof that $\tilde{\Phi}$ is indeed a functor. $\square$

4.2.2 Characterization as a left Kan extension

The idea that we want to visit is to replace the fully shifted local transition function by only the usual local transition function to generate the coarse transition functor as a left Kan extension. Unfortunately any transition function outputting a single state from each local configuration is doomed to failure. Indeed, a singleton configuration defined at any position $g \in G$ is isomorphic to the singleton configuration with same state at any other position $g' \in G$. It follows that, categorically speaking, such a local transition function does not provide any positioning information of the resulting states. It is however possible to save the relative positioning between the states by relating pairs of resulting states from pairs of local configurations. Indeed, even when two local configurations are separated by some element $b \in G$, so is their pair of results. Moreover, it is not necessary to take all pairs of configurations. It is enough to consider a generating set of the group, all other relative positioning being obtained transitivity from them.

Let us make these ideas precise and fix once and for all a generating set $B$ of $G$, that is, a subset of $G$ such that any element of $G$ can be expressed as a finite combination of elements of $B$ and their inverses under the group operation. For each a relative position $b \in B$, we consider the two positions $1$ and $b$, that we call centers, and take all configurations defined on the union of the neighborhood of the centers, that is, with support $1.N \cup b.N$. We also consider $1 \in B$. Indeed, with $b = 1$, the two centers equalize and local configurations are automatically included. The interplay between local configurations and pairs of local configurations is essential and arise from arrows between them. An arrow between two such configurations is an arrow from $\text{Conf}$ satisfying that centers are sent to centers.

**Definition 52** We defined $\text{Loc}$ to be the category with as objects the configurations $\text{Ob}_{\text{Loc}} = \bigcup_{b \in B} \{b\} \times Q_{1,N/b,N} = \bigcup_{b \in B} \{b\} \times Q_{1}^{1,N/b,N}$ defined on pair of neighborhoods, and, for any two such configurations $(b, c)$ and $(b', c')$, an arrow $g : (b, c) \to (b', c')$ is the data of $g : c \to c' \in \text{Conf}$ such that $g \cdot \{1, b\} \subseteq \{1, b'\}$.

Composition in $\text{Loc}$ is inherited from $\text{Conf}$. For any object $(b, c) \in \text{Loc}$, the arrow $1 : (b, c) \to (b, c)$ acts as the identity.

The relation $g \cdot \{1, b\} \subseteq \{1, b'\}$ encodes the intuition of centers being sent to centers. Indeed, consider a center $p \in \{1, b\}$ of the configuration $(b, c)$. Since, the arrow $g : c \to c'$ means that $c \leq c' \bowtie g$, we have $c(p) = (c' \bowtie g)(p) = c'(g \cdot p)$. So we expect $g \cdot p$ to be a center of $(b', c')$: either $1$ or $b'$. Now, there are not many possible triples $(b, b', g \in G$ satisfying $g \cdot \{1, b\} \subseteq \{1, b'\}$ and it is useful to enumerate the different cases:

1. Identities $(g.1 = 1$ and $g.b = b')$: in this case, we have $g = 1$ and $b = b'$ implying that $c = c'$ and $g = id_c = id_{(b,c)}$ is the identity arrow.

2. Other isomorphisms $(g.1 = b'$ and $g.b = 1$): this case, for which we have $g = b' = b^{-1}$, holds when $B$ contains $b$ and its inverse. The two configurations $c$ and $c'$ describe the exact same local states but where centers are reversed. The two objects $(b, c)$ and $(b', c')$ in $\text{Loc}$ are then isomorphic. This case is not useful and may be dropped by considering $B$ without inverse.

3. Local configuration around center 1 $(g.1 = 1$ and $g.b = 1$): in this case, $b = 1$ which means that $c$ is a local configuration. Moreover, we have $g = 1 : c \to c'$ which identifies the occurrence of $c$ on the first center 1 of $c'$.

4. Local configuration around center $b$ $(g.1 = b'$ and $g.b = b')$: in this case, $b = 1$ which also means that $c$ is
a local configuration. But now, we have \( g = b' : c \to c' \) which identifies the occurrence of \( c \) on the second center \( b' \) of \( c' \).

**Proposition 53** \textit{Loc} is indeed a category.

**Proof** For any \((b, c) \in \text{Loc}\) the arrow \( \text{id}_{(b, c)} = 1 \) is well defined as \( 1 \cdot \{1, b\} = \{1, b\} \). The composition operation is well defined. Indeed, given \( f : (b, c) \to (b', c') \) and \( g : (b', c') \to (b'', c'') \), we have:

\[
\begin{align*}
& f \cdot \{1, b\} \subseteq \{1, b'\} \land g \cdot \{1, b'\} \subseteq \{1, b''\} \\
\iff & (g \cdot f) \cdot \{1, b\} \subseteq g \cdot \{1, b'\} \land g \cdot \{1, b'\} \subseteq \{1, b''\} \\
\Rightarrow & (g \cdot f) \cdot \{1, b\} \subseteq \{1, b''\},
\end{align*}
\]

which implies that \( g \circ f = g \cdot f \) is an arrow of \textit{Loc}.

Associativity of the composition and neutrality of the identities are inherited from \textit{Conf} and already proved in Proposition 44. \qed

**Definition 54** We consider the functor \( \pi_2 : \text{Loc} \to \text{Conf}^{\ast} \) defined by \( \pi_2(b, c) = c \) on objects and \( \pi_2(g) = g \) on arrows.

**Proposition 55** \( \pi_2 \) is indeed a functor.

**Proof** For any object \((b, c) \in \text{Loc}\), \( c \) is by definition an object of \textit{Conf} as \( 1 \cdot N \subseteq |c| \) and \( b \cdot N \subseteq |c| \). Given any arrow \( g : (b, c) \to (b', c') \) we observe that \( g : c \to c' \) is an arrow of \textit{Conf} and \( c \) and \( c' \) are object of \textit{Conf}*. Then by the fact that \textit{Conf} is a full subcategory of \textit{Conf} it follows that \( g \) is an arrow of \textit{Conf}*. As the mapping \( \pi_2 \) is the identity on arrows, compositions and identities are trivially respected. \qed

**Definition 56** The local transition functor \( \hat{\delta} : \text{Loc} \to \text{Conf} \) is defined on any objects \((b, c)\) by \(|\hat{\delta}(b, c)| = \{1, b\}\) and \( \hat{\delta}(b, c)(p) = \delta(c \map p|N) \), and on any arrows \( g : (b, c) \to (b', c') \) by \( \hat{\delta}(g) = g \).

**Proposition 57** The local transition functor \( \hat{\delta} \) is indeed a functor.

**Proof** First \( \hat{\delta} \) is well defined on objects since for any \((b, c) \in \text{Loc}\), \( \hat{\delta}(b, c) \) is in \( Q^{(1, b)} \) which is a subset of \( \text{Ob}_{\text{Conf}} \).

Then, for any arrow \( g : (b, c) \to (b', c') \in \text{Loc} \), we need to show that \( \hat{\delta}(g) \) is an arrow of \textit{Conf} from \( \hat{\delta}(b, c) \) to \( \hat{\delta}(b', c') \). By Definition 43 of \textit{Conf}, it rewrites to \( \hat{\delta}(b, c) \leq \delta(b', c') \wedge g \). We first show that \(|\delta(b, c)| \leq |\delta(b', c') \wedge g|\). For coincidence, consider \( |\hat{\delta}(b, c)| \):

\[
|\hat{\delta}(b, c)| = |\Phi(c)| = \text{int} (|c|) = \{g \in G \mid g \cdot N \subseteq |c|\} = \{g \in G \mid g \cdot N \subseteq \{1, b\} \cdot N\} \supseteq \{1, b\} = |\hat{\delta}(b, c)|
\]

For coincidence, consider \( g \in |\hat{\delta}(b, c)| \):

\[
\hat{\delta}(b, c)(g) = \delta(c \map g|N) = \Phi(c)(g) = (\Phi \circ \pi_2)(b, c)(g).
\]

It remains to show the naturality condition on \( \eta \) which means to check that for any \( g : (b, c) \to (b', c') \in \text{Loc} \), we have \( \eta_{(b', c')} \circ \hat{\delta}(g) = (\Phi \circ \pi_2)(g) \circ \eta_{(b, c)} \). It is simply given by \( \eta_{(b', c')} \circ \hat{\delta}(g) = 1 \cdot g = g = \Phi \circ \pi_2(g) \circ \eta_{(b, c)} = g \cdot 1 = g \). \qed

The following lemma will be used in the proof of Theorem 61.

\( \square \) Springer
Lemma 60 Given a set D and map \( f : B \times G \to D \). If for any \((b, g) \in B \times G\), we have \( f(1, g) = f(b, g) = f(1, g \cdot b) \) then \( f \) is constant.

Proof Since for any \( b \) and \( g, f(b, g) = f(1, g) \), it is enough to show that for any \( g, f(1, g) = f(1, 1) \) to show that \( f \) is a constant function. Since \( B \) is a generating set of \( G, g = b_1 \ldots b_k \) for some natural integer \( k \) and \( b_i \in B \cup B^{-1} \). In this setting, we define \( h_i = b_1 \ldots b_i \) for \( 0 \leq i \leq k \). We now show that \( f(1, h_{i+1}) = f(1, h_i) \) for \( 0 < i \leq k \). For \( b_{i+1} \in B, f(1, h_{i+1}) = f(h_i, b_{i+1}) = f(1, h_i) \) by assumptions on \( f \). If \( b_{i+1} \in B^{-1}, f(1, h_{i+1}) = f(b_{i+1}^{-1}, h_i) = f(h_i, b_{i+1}^{-1}) \) by using the assumptions on \( f \). Then we get \( f(1, h_{i+1} \cdot b_{i+1}^{-1}) = f(1, h_i \cdot b_{i+1} \cdot b_{i+1}^{-1}) = f(1, h_i) \). By transitivity, we obtain \( f(1, g) = f(1, h_k) = f(1, 1) \). □

Theorem 61 \((\hat{\Phi}, \eta)\) is the left Kan extension of \( \hat{\Phi} \) along \( \pi_2 \).

Proof Taking any other \( \hat{\Psi} : \text{Conf}^* \to \text{Conf} \) and \( \rho : \hat{\Phi} \to \pi_2 \) we need to show that there exists a unique natural transformation \( \hat{\lambda} : \hat{\Phi} \to \hat{\Psi} \) such that for any \((b, l) \in \text{Loc}\) we have \( \hat{\lambda}_c \circ \eta_{(b, l)} = \rho_{(b, l)} \).

Given some \( c \in \text{Conf}^* \), observe that for any \((b, l) \in \text{Loc}\) and \( g : l \to c \) in \( \text{Conf}^* \), \( \hat{\lambda}_c \) must be such that:

\[
\begin{align*}
\hat{\lambda}_c & : \hat{\Phi}(g) = \hat{\Psi}(g) \circ \hat{\lambda}_l \\
\hat{\lambda}_c & : \hat{\Phi}(g) \circ \eta_{(b, l)} = \hat{\Psi}(g) \circ \hat{\lambda}_l \circ \eta_{(b, l)} \\
\hat{\lambda}_c & : \hat{\Phi}(g) \circ \eta_{(b, l)} = \hat{\Psi}(g) \circ \rho_{(b, g)}
\end{align*}
\]

Under these constraints we let \( \hat{\lambda}_c^{(b, g)} \) to be the unique group element that satisfies Equation (1), i.e., \( \hat{\lambda}_c^{(b, g)} := \hat{\Psi}(g) \circ \rho_{(b, g)} \cdot \eta_{(b, g)}^{-1} \cdot \hat{\Phi}(g)^{-1} \), for any \((b, l) \in \text{Loc}\) and \( g : l \to c \) in \( \text{Conf}^* \). Observe that there must be at least one such \((b, l)\). Indeed as \( c \in \text{Conf}^* \) there must be some \( s \) such that \( s \cdot N \subseteq |c| \) so one can take \( b = 1 \) and \( l = c \cdot s \cdot 1 |N| \).

Let us show that the value of \( \hat{\lambda}_c^{(b, g)} \) does not depend on the choice of \((b, l)\) and \( g \). We start with the special case where \( c \in \Omega^2 \) is a global configuration.

Notice that the triples \((b, l, g)\) with \((b, l) \in \text{Loc}\) and \( g : l \to c \), are in bijection with the pair \((b, g) \in B \times G\), since for such a given pair the unique candidate is \( l := c \cdot g \cdot |N| \cdot B \cdot N \).

We are then able to define a function of \( B \times G \to G \) mapping any \((b, g)\) to \( \hat{\lambda}_c^{(b, g)} \). We show that all the \( \hat{\lambda}_c^{(b, g)} \) are equal by showing that this function is constant. For, we use Lemma 60 which simply requires that for any \((b, g) \in B \times G\):

\[
\lambda_c^{(b, g)(b, h)} = \lambda_c^{(b, l)(b, h)}
\]

where \( h_1 := 1 : (1, l |N|) \to (b, l) \) and \( h_b := b : (1, l |b| |N|) \to (b, l) \). The two required equalities are derived as follows.

\[
\begin{align*}
\lambda_c^{(b, l)(b, h)} & = \hat{\Psi}(g \circ h_1) \cdot \rho_{(1, l |N|)} \cdot \eta_{(1, l |N|)}^{-1} \cdot \hat{\Phi}(g \circ h_1)^{-1} \\
& = \hat{\Psi}(g) \cdot \hat{\Psi}(h_1) \cdot \rho_{(1, l |N|)} \cdot \eta_{(1, l |N|)}^{-1} \cdot \Phi(h_1)^{-1} \cdot \Phi(g)^{-1} \\
& = \hat{\Psi}(g) \cdot \rho_{(b, g)} \cdot \delta(h_1) \cdot \delta(h_1)^{-1} \cdot \eta_{(b, g)}^{-1} \cdot \Phi(g)^{-1} \\
& = \hat{\Psi}(g) \cdot \rho_{(b, g)} \cdot \eta_{(b, g)}^{-1} \cdot \Phi(g)^{-1} \\
& = \lambda_c^{(b, l)}
\end{align*}
\]

which concludes the proof that the value of \( \lambda_c^{(b, g)} \) does not depend on the choice of \((b, l)\) and \( g \). Consequently, we are able to define \( \lambda_c := \lambda_c^{(b, l)} \). To finally get the component arrows, it remains to show that \( \lambda_c \in \text{Hom}_{\text{Conf}}(\hat{\Phi}(c), \hat{\Psi}(c)) \), i.e., \( \hat{\Phi}(c) \triangleright \hat{\lambda}_c \). Take \( g \in |\hat{\Phi}(c)| \) and consider \((1, l_g) \in \text{Loc}\) with \( l_g := c \cdot g \cdot |N| \).

We obviously have \( l_g \leq c \cdot g \), which means that we also have an arrow \( g : l_g \to c \). By the existence of the arrow \( \hat{\Psi}(g) \circ \rho_{(1, l_g)} : \delta(1, l_g) \to \hat{\Psi}(l_g) \to \hat{\Psi}(c) \), we have:

\[
\lambda_c^{(1, l_g)(1, l_h)} = \lambda_c^{(1, l_g)(1, l_h)}
\]
\[
\delta(1, l_g) \preceq (\Psi(c) \cdot \bar{\Phi}(g) \cdot \rho(l_0)) \\
\iff \delta(1, l_g) \preceq (\bar{\Phi}(g) \cdot \rho(l_0) \cdot g^{-1} \cdot g) \\
\iff \delta(1, l_g) \preceq (\bar{\Phi}(g) \cdot \rho(l_0) \cdot g^{-1} \cdot g) \\
\iff \delta(1, l_g) \preceq (\bar{\Phi}(g) \cdot \rho(l_0) \cdot \eta(1)_{\bar{\Phi}}^{-1} \cdot \bar{\Phi}(g)^{-1} \cdot g) \\
\iff \delta(1, l_g) \preceq (\Psi(c) \cdot \bar{\lambda}_c \cdot g) \\
\iff g \in (\Psi(c) \cdot \bar{\lambda}_c) \land (\bar{\Phi}(c) \cdot \bar{\lambda}_c) = \delta(l_g)
\]

reminding that \(1 \in [\delta(1, l_g)]\). But \(\bar{\Phi}(c)(g) = \Phi(c)(g) = \delta(l_g)\).

We get \(\bar{\Phi}(c)(g) = (\Psi(c) \cdot \bar{\lambda}_c)(g)\) so \(\bar{\lambda}_c\) is a well defined arrow.

We then show that for any \((b, l) \in \mathbf{Loc}\), we have \(\lambda_t \circ \eta_{b, b} = \rho_{b, b} \). Taking the identity arrow \(id_l : l \rightarrow l\), it gives:
\[
\lambda_t \cdot \eta_{b, b} = (b, l) \cdot \eta_{b, b} = \Psi(id_l) \cdot \rho_{b, b} \cdot \eta_{b, b}^{-1} \cdot \bar{\Phi}(id_l)^{-1} \cdot \eta_{b, b} = \rho_{b, b} \cdot \eta_{b, b}^{-1} \cdot \eta_{b, b} = \rho_{b, b}
\]

Finally we show the naturality of \(\bar{\lambda}_c\), i.e., for any \(g : c \rightarrow c' \in \mathbf{Conf}\) we have \(\bar{\Psi}(g) \circ \bar{\lambda}_c = \bar{\lambda}_c' \circ \bar{\Phi}(g)\). To see this, take any \((1, l) \in \mathbf{Loc}\) such that there is \(i : l \rightarrow c\) and observe that:
\[
\lambda_t \cdot \bar{\Phi}(g) = \bar{\Psi}(g \circ i) \cdot \rho_{(1, b)} \cdot \eta_{(1, b)}^{-1} \cdot \bar{\Phi}(g \circ i) \cdot \bar{\Phi}(g)^{-1} = \bar{\Psi}(g) \cdot \rho_{(1, b)} \cdot \eta_{(1, b)}^{-1} \cdot \bar{\Phi}(g)^{-1} = \bar{\Psi}(g) \cdot \bar{\Phi}_c
\]

which shows that \(\bar{\lambda}_c\) is natural.

Uniqueness comes from Eq. (1) which defines uniquely the component arrows.

\[\square\]

### 4.3 Kan extensions are "up to isomorphisms"

In the case of poset Kan extensions, such an extension is unique when it exists. This is not the case for the category Kan extensions that we consider due to the degree of freedom of the absolute positioning of configurations. In particular a cellular automaton and its shifted versions lead to isomorphic coarse transition functors, so they are Kan extensions of the same data as we now show.

**Definition 62** For any \(s \in G\), we call the \(s\)-shifted cellular automaton as the cellular automaton having neighborhood \(N_s = s \cdot N\), the same set of states \(Q\), and the local transition function \(\delta_s : Q^N \rightarrow Q\) defined as \(\delta_s(c) := \delta(c \cdot s)\).

**Proposition 63** For any \(c \in Q^N_s\), \(\delta_s(c)\) is well defined.

**Proof** Since \(\delta : Q^N \rightarrow Q\), we simply need to show that \([c \cdot s] = N\):
\[
|c \cdot s| = \{h \in G \mid s \cdot h \in [c]\} = \{h \in G \mid s \cdot h \in s \cdot N\} = N
\]

\[\square\]

**Definition 64** Given any \(s \in G\), the \(s\)-shifted coarse transition function \(\Phi_s : \mathbf{Conf} \rightarrow \mathbf{Conf}\) is defined for all \(c \in \mathbf{Conf}\) as \(|\Phi_s(c)| = \text{int}_s(|c|) = \{g \in G \mid g \cdot N_s \subseteq \|c\|\}\) and \(\Phi_s(c)(g) = \delta_s(c \cdot g | N_s)\).

**Proposition 65** For any \(s \in G\) such that \(s \cdot g = g \cdot s\) for all \(g \in G\), the shifted coarse transition functor \(\bar{\Phi}_s\) is isomorphic to \(\bar{\Phi}\), i.e., there exists a natural transformation \(\gamma : \bar{\Phi}_s \Rightarrow \bar{\Phi}\) such that for any \(c \in \mathbf{Conf}\) there is some arrow \(g : \bar{\Phi}(c) \rightarrow \bar{\Phi}_s(c)\) such that \(s \circ g = id_{\mathbf{Conf}}\) and \(g \circ \gamma_s = id_{\bar{\Phi}(c)}\).

**Proof** Let \(\gamma_s = s\) for any \(c \in \mathbf{Conf}\). First we show that \(\gamma_s\) is an arrow from \(\bar{\Phi}_s(c)\) to \(\bar{\Phi}(c)\), i.e., \(\gamma_s(c) \preceq \bar{\Phi}(c)\cdot s\):
\[
\forall g \in [\bar{\Phi}_s(c)] \cdot g \in [\bar{\Phi}(c)\cdot s] \land \bar{\Phi}_s(c)(g) = (\bar{\Phi}(c)\cdot s)(g)
\]

line 2 is given by Definitions 64 and 6, line 3 by Definitions 62 and 19 and line 4 by Propositions 20, 8. The last line is true as \(g \cdot s = s \cdot g\).

In fact as \(g \in [\bar{\Phi}_s(c)]\) is equivalent to \(g \cdot s \cdot N_s \subseteq \|c\|\) which is also equivalent to \(g \in [\bar{\Phi}(c)\cdot s]\) we have \(\bar{\Phi}_s(c) = (\bar{\Phi}(c)\cdot s)\). It follows that \(s \cdot \bar{\Phi}_s(c) \rightarrow \bar{\Phi}_s(c)\) is a valid arrow. But we can write equivalently \(\bar{\Phi}_s(c)\cdot s^{-1} = \bar{\Phi}_s(c)\) which gives that \(s^{-1} : \bar{\Phi}_s(c) \rightarrow \bar{\Phi}_s(c)\) is a valid arrow. Then by \(s \circ s^{-1} = s \cdot s^{-1} = 1\) and \(s^{-1} \circ s = s^{-1} \cdot s = 1\) we get that \(s\) is an isomorphism.

Finally we show the naturality of \(\gamma_s\), i.e., \(\bar{\Phi}(f) \circ \gamma_s = \gamma_s \circ \bar{\Phi}_s(f)\) for any \(f : c \rightarrow c'\):
\[
\bar{\Phi}(f) \circ \gamma_s = f \cdot s = s \cdot f = \gamma_s \circ \bar{\Phi}_s(f)
\]

It is true as line 1 holds by Definition 50 and by definition of \(\gamma_s\), line 2 as \(f \cdot s = s \cdot f\) and line 3 holds by definition of \(\gamma_s\) and Definition 50.
5 Final discussion

In this work, we propose to revisit the relation local/global in cellular automata through the perspective of arbitrary partial configurations. The relation itself is shown to be expressed in terms of Kan extensions, extending the local behavior specified by the transition table to any partial configurations.

A first construction, restricted to posets only, shows that two particular extensions, the coarse and fine monotonic transition functions $\phi$ and $\Phi$, correspond respectively to some left and right Kan extensions. There are additional simple structural facts to note about the monotonic functions considered. The first one is that the shift action on partial configurations, as given in Definition 6, is the right Kan extension of the shift action on global configurations, as given in Definition 2. Another one is that the coarse monotonic transition function $\Phi$ is always smaller than the fine transition function $\phi$ $(\Phi \Rightarrow \phi)$, hence the names of these transition functions, coarse and fine. In fact, any transition function $f : \text{Conf} \rightarrow \text{Conf}$ such that $\overline{\delta} \Rightarrow f \circ \pi_2$ is necessarily such that $\Phi \Rightarrow f \Rightarrow \phi$. This means that any function respecting at least the local transition of the cellular automaton is in between the coarse and fine transition functions. These functions have a structure of poset with $\Phi$ and $\phi$ as extremal extensions. This shows, in some sense, the efficiency of the simple constraints of monotonicity and $\overline{\delta} \Rightarrow f \circ \pi_2$ for describing extensions. In this first formal development, the single local behavior $\delta$ is explicitly “copied” on all $g \in G$ to obtain $\overline{\delta}$. It is readily possible to put a different behavior on each $g \in G$, with no real modification to the proofs. The statements are therefore valid for non-uniform cellular automata and automata networks. Notice that any constraint on the finiteness of the number of states is not mentioned either.

A second construction is proposed to prevent the use of the highly redundant “fully shifted local transition function” $\overline{\delta}$. It consists in using the shift operation to relate configurations into a category of configurations instead of a poset of configurations. The former is very similar to the poset, except that the yes/no question “is this configuration a subconfiguration of this other one?” is replaced by the open-ended question “where does this configuration appear in this other one?” (Maignan et al. 2015). The constructions in the poset setting can then be revisited into this new setting. We focused our study on lifting the case of the coarse monotonic transition function and get the coarse transition functor instead. We leave the definition of a fine transition functor and the subsequent results for future works. Indeed, the lifting is pretty straightforward but some important differences emerge and need to be taken carefully. They are essentially due to the degree of freedom gained by the possibility to relate previously not comparable configurations, thanks to the shift operation. For example, the reader might have noticed the use of $\text{Conf}^\star$ instead of $\text{Conf}$ which restricts the extension to super local configurations only. Indeed, for configurations with an empty interior, the shift operation allows in some case many possibilities in positioning their results breaking the expected uniqueness property. Another difference is that Kan extensions, as any categorical construction, are defined up to isomorphism, so that they do not allow to distinguish between isomorphic constructions. In our case, we have shown that, in some conditions, the coarse transition functors of two shifted automata are isomorphic. In other words, the given construction is not able to distinguish between some different cellular automata.

These limitations are not surprising. They are even expected in some way. The poset case is definitely the appropriate setting for describing cellular automata in their strict definitions, and the proposed result, with small modifications, is closely related to the Curtis–Hedlund theorem. Indeed, the “fully shifted local transition function” $\overline{\delta}$ is clearly an implementation of the required shift-equivariance condition, and if one restores the constraint of a finite set of states, one can see that the poset of finite support configurations is a “generating” part of the poset of open subsets of the product topology. In this case, the fine transition function $\phi$ can be viewed as encoding an important part of the topological behavior of the global transition function $\Delta$ (Ceccherini-Silberstein and Coornaert 2010; Hedlund 1969).

On the other hand, the transition to the categorical case has been thought to be compatible with generalizations of cellular automata to more arbitrary or dynamical spaces. In these cases, there is no convenient absolute positioning systems and it is more natural to think in terms of occurrences of a configuration in another as we exactly did in $\text{Conf}$. The direct consequence is the presence of isomorphisms making shifted configurations not distinguishable. This new setting goes out of the context of cellular automata strictly speaking, but is more suitable for working with cellular automata modulo shift. From this perspective, the categorical construction is to be considered in the light of global transformations (Fernandez et al. 2021, 2019; Maignan et al. 2015). These generalizations are a proposition adapting the main ingredients of cellular automata where space is abstracted away and presented as an arbitrary category similar to $\text{Conf}$. The local/global relationship is described by a decomposition/recomposition process which is efficiently presented as a Kan extension. In particular, (Fernandez et al. 2019) applies this generic construction to a category of finite words allowing to recover deterministic Lindenmayer systems which are...
parallel rewriting of strings. Another example of dynamical spaces can be found in (Maignan et al. 2015) where global transformations are shown to capture a large family of synchronous evolving graphs (providing an interesting alternative to the causal graph dynamics of (Arrighi and Dowek 2012)). In this case, the considered category has graphs as objects and structural graph inclusions as arrows. Lastly, evolving higher-order dimensional structures such as abstract cell complexes are captured in (Maignan et al. 2015). An example of mesh subdivision is given illustrating the decomposition/recomposition mechanism. In this context, a (global) configuration is the data of a triangular mesh; it is decomposed into triangles which are independently subdivided obeying to the system local transition rules; the local subdivided triangles are then merged all together to get the final global result. The reader is invited to pay attention that the proofs given in the present work, especially for Theorem 61, also follow the same decomposition/recomposition process involved in the local/global definition of cellular automata.

To finish, let us mention an important aspect of some Kan extensions considered here and in the other papers (Fernandez et al. 2021, 2019; Maignan et al. 2015). They have the property to be pointwise. Intuitively, this means that they can be computed “algorithmically” using simple building blocks. This formulation in terms of building blocks is completely equivalent and is the one used in the other papers, firstly because it is via these building blocks that the authors discovered these links between spatially-extended dynamical systems and category theory, and secondly because this formulation is closer to the software implementations of the considered models. In fact, it is possible to have an implementation completely generic over the particular kind of space considered. The specification of such an algorithm can be found in (Fernandez et al. 2021) for the considered large family of evolving graphs.

References

Arrighi P, Dowek G (2012) Causal graph dynamics. In: Artur C, Kurt M, Andrew MP, Roger W (eds) 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. Springer, pp 54–66. Springer

Banâtre J-P, Fradet P, Le MD (2000) Gamma and the chemical reaction model: fifteen years after. In: Workshop on membrane computing. Springer. pp 17–44

Berry Gerard, Boudol G (1992) The chemical abstract machine. Theor Comput Sci 96(1):217–248

Ceccherini-Silberstein T, Coornaert M (2010) Cellular automata and groups. Springer Science & Business Media

Fernandez A, Maignan L, Spicher A (2019) Lindenmayer systems and global transformations. In: Ian M and Shinosuke S (eds) 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. Springer, pp 65–78

Fernandez A, Maignan L, Spicher A (2021) Accretive computation of global transformations. In: International conference on relational and algebraic methods in computer science. Springer, pp 159–175

Hedlund GA (1969) Endomorphisms and automorphisms of the shift dynamical system. Math Syst Theor 3(4):320–375

MacLane S (2013) Categories for the working mathematician. In: Graduate texts in mathematics. Springer, New York

Maignan L, Spicher A (2015) Global graph transformations. In: Detlef P (eds) 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, pp 34–49. CEUR-WS.org

Pa˘un Gheorghe (2001) From cells to computers: computing with membranes (p systems). Biosystems 59(3):139–158

Rozenberg G, Salomaa A (2012) Lindenmayer systems: impacts on theoretical computer science, computer graphics, and developmental biology. Springer Science & Business Media

Spicher A, Giavitto J-L (2017) Interaction-based programming in mgs. In: Advances in unconventional computing. Springer, pp 305–342

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.