Universiteit Leiden

Master Computer Science

A coalgebraic modal logic for cellular automata

Name:	Lulof Pirée
Student ID:	s3406512
Date:	15/07/2024
Specialisation:	Artificial Intelligence
1st supervisor:	Dr. Henning Basold
2nd supervisor:	Dr. Chase Ford

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Einsteinweg 55
2333 CC Leiden
The Netherlands

Abstract

Dynamic patterns of states in the evolution of a cellular automaton can be described in the simple syntax of our new modal logic. Not only can the logic express local properties such as the local state-transition rule, but also global properties such as periodicity and nilpotency. We present the semantics of the logic via a novel coalgebraic model of cellular automata that explicitly separates cells from connections between them. This model is sufficiently general to describe any graph shape, and generalises to non-uniform cellular automata in which different cells may admit different local rules. Finally, we define a coalgebraic definition of similarity that includes a map between the parameters of the type functor. A generalised Hennessey-Milner theorem proves that two cellular automata are similar if and only if every modal logic formula valid on the one can be translated into a formula valid on the other.

Contents

1 Introduction **3**
1.1 Related work . . . 5

2 Preliminaries **6**
2.1 Sets and functions . . . 7
2.2 Cellular automata and monoids . . . 8
2.3 Category theory . . . 11
2.4 Modal logic . . . 13

3 Uniform cellular automata **15**
3.1 Graph as coalgebras . . . 15
3.2 Extending graphs with local rules . . . 19
3.3 Uniform cellular automata as coalgebras . . . 19
3.4 Modelling irregular and non-uniform CA . . . 25
3.5 Summary of uniform CA . . . 27

4 General cellular automata **27**
4.1 Classes of general CA . . . 27
4.2 General CA functor . . . 34
4.3 Neighbourhood- and extension-uniform CA . . . 36
4.3.1 The local rule as a natural transformation . . . 37
4.4 Summary of general CAs . . . 38

5 Coalgebra homomorphisms and behavioural equivalence **38**
5.1 Characterising coalgebra homomorphisms . . . 39
5.1.1 First property: commutativity with cowriter-comonad coalgebras . . . 39
5.1.2 Second property: extension of local rules . . . 41
5.2 Illustration of behavioural equivalence . . . 42
5.3 The terminal coalgebra . . . 43
5.3.1 Coalgebra contractions . . . 44
5.3.2 Construction of the terminal coalgebra . . . 51
5.3.3 Morphisms between embeddings . . . 52
5.3.4 Cardinality of union of contractions . . . 56
5.4 Summary of coalgebra homomorphisms and behavioural equivalence . . . 58

6 Logic for computation traces **59**
6.1 Space modalities . . . 59
6.1.1 Grammar . . . 60
6.1.2 Semantics . . . 60

6.1.3 Expressivity of space modalities . . . 61
6.2 Time modalities . . . 62
6.2.1 Expressivity of time modalities . . . 63
6.3 Global variables and uniform substitution . . . 64
6.4 Logical reasoning rules . . . 67
6.5 Summary of basic modal logic . . . 70

7 Logic extension: infinitary disjunctions 71
7.1 Intuition and motivation . . . 71
7.2 Formal definitions of infinitary disjunctions . . . 72
7.3 Examples of expressible properties . . . 74
7.4 Summary of infinitary disjunctions . . . 79

8 Correspondence semantical and logical bisimulation 79
8.1 Examples of similarity . . . 80
8.1.1 A basic example of similarity . . . 80
8.1.2 Motivation extending similarity with monoid homomorphisms . . . 80
8.1.3 Motivation extending similarity with state mappings . . . 82
8.2 Basic Hennessy-Milner theorem . . . 82
8.3 Generalised Hennessy-Milner theorem . . . 88
8.3.1 General CA morphisms . . . 88
8.3.2 Generalised notions of similarity . . . 89
8.3.3 Generalised Hennessy-Milner theorem . . . 91
8.3.4 Example applications . . . 95
8.4 Summary of the Hennessy-Milner theorem . . . 97

9 Concluding remarks 98
9.1 Discussion of the coalgebaic model . . . 98
9.2 Discussion of the new modal logic . . . 98
9.3 Conclusion . . . 99

References 100

1 Introduction

Cells in a human body, particles in a gas cloud, ants in an ant-hill, components in electronic circuits; the world is full of systems consisting of a collection of simple objects (cells, particles, ants, components) that interact only locally with similar objects in their surrounding. Cellular automata (CA) are computational models for studying such systems mathematically and/or simulating them on a computer. In practise, CA are employed in the empirical sciences to for example biological [10] and physical systems [32][33], but also as tools in areas such as game design [47][17][44], music generation [3] and in and cryptography [28][49][27] [46]. In the theory of computation, CA are a very general computational model: many other models such

such as Turing Machines [42][24], have be described as specific CA.

A CA consists of a graph where every vertex (usually called a *cell*) is inhabited by a deterministic automaton. These cells hold a single symbol, their *state*, and in discrete time steps they synchronously take their neighbour's states as inputs and update their own state according to a deterministic *local rule*. This defines a global transition function between assignments of states to the cells (*configurations*). From a given initial configuration, one can recursively compute the *trace sequence* of successor configurations using this global function. Despite the simplicity of individual cells, consisting of no more than a single state and a deterministic local rule, the global trace can exhibit complex behaviour.

Local spatial and/or temporal patterns in the trace of CA are often of great importance in their analysis and design. Such patterns are, for example, data structures or signals between isolated groups of actively interacting cells. Currently, there is a deficit in the supply of universal tools to describe, present, compare and reason about such patterns. Visualisations are commonly employed, but are informal and describe only limited portions of space and time. Furthermore, some graphs (such as four-dimensional grids) can be visualised. The aim of this study is to provide a symbolic logic capable of describing spatio-temporal patterns with a uniform syntax for all graphs.

Modal logics are logical languages that have successfully been employed to describe and reason about local properties of various computational models, such as labelled transition systems [12]. Essentially, modal logic is propositional (Boolean) logic that uses a coloured graph of *worlds*, which are assignments of Boolean values to variables, instead of a single such assignment. Formulas can reason about truths in a given world, but also evaluate subformulas in worlds related via a coloured edge. It is this graph-based nature that makes modal logic an inviting choice for a logic of CA: spatially, CA have cells arranged in a graph, and temporally, CA represent their computation via their trace sequences. Hence the research question of the present work:

> **Research question:** *Is it possible to define a modal logic on CA, capable of describing spatial and temporal relations of cells' states in a trace sequence?*

Many dynamical systems have been be modelled in the form of *coalgebras* [39], a concept from category theory that has been proposed to be a universal framework for modal logic [6][23]. In particular, here exists a standard framework of *predicate-lifting* developed by Pattinson [31][30], Schröder [41] and Klin [22] for automatically generating a modal logic with an expressivity guarantee for coalgebras.

Hence a natural starting point for the present study was to model CA coalgebraically as to employ this framework. This line of thought led to a natural generalisation of CA to *non-uniform* cellular automata, in which distinct cells are allowed to have different local rules. However, we show that the predicate-lifting framework would generate a modal logic that is only capable of describing static properties of the graph structure of a CA, and not of properties of computation traces.

Fortunately, the present coalgebraic model provided a suitable foundation for the semantics of a tailor made modal logic that is capable of describing patterns in traces. The logic is especially expressive when extending it with infinitary disjunctions; in some cases can even describe certain global asymptotic properties of traces.

Finally, we define a notion of *mutual similarity* on cells in two CA, based on the ability to simulate each other. In a Hennessy-Milner style theorem we establish that CA are mutually similar if and only if they have a similar logic. In fact, we prove the stronger statement that a cell x can be simulated by another cell if and only if x's logic can be embedded in y's logic, using a translation of incompatible logical symbols.

Summary of contributions The contributions of this thesis can be summarised as follows:

1. A new coalgebraic model of uniform and non-uniform cellular automata based on a **Set**-valued comonad, which explicitly separates cells from connections between cells. We propose different classes of non-uniform CA that arise as full subcategories of our coalgebras.

2. When using the logic generated by the predicate lifting framework, two cells would have the same logic if they are *behaviourally equivalent*. We show how CA can be *contracted* to the minimal CA to which it is behaviourally equivalent, and how all contractions can be embedded in a *terminal coalgebra*. The outcome of this study is that many CA are BE, even when their trace sequences on the same initial configuration are different. Hence the predicate lifting's logic would be unable to describe the details of the trace sequences, which our design goal requires.

3. A tailor made modal logic for describing spatio-temporal patterns in traces of non-uniform CA: an extension of propositional logic with unary modalities for connections between cells and for timesteps in a trace sequence. We show how it can be extended with infinitary disjunctions, and give examples properties of CA that the logic can express.

4. A Hennessey-Milner style theorem establishing the equivalence of semantical simulation and logical simulation.

Thesis overview In Section 2 we gather the required background concepts and we set notational conventions used in this thesis. The main ideas behind our coalgebraic model are introduced in Section 3, which also proves a correspondence between the classical and coalgebraic definitions of CA. Thereafter, Section 4 refines those ideas, and simultaneously generalises them to non-uniform CA; in particular it shows how definitions of two subclasses of non-uniform CA can naturally be formulated in terms of coalgebras. Section 5 analyses morphisms between our coalgebras, and shows that the standard framework for coalgebraic modal logic does not provide the desired logic. We explain the basic version of our manually designed logic in Section 6, and extend it with infinitary disjunctions in Section 7. The latter section also showcases examples of what the logic can express. In Section 8 we define simulations and logical simulations between cells, and prove their relation in a Hennessy-Milner style theorem The concluding remarks in Section 9.3 finish the thesis.

1.1 Related work

Coalgebraic modal logic The semantics of various modal logics can be defined in terms of coalgebras, and general results for coalgebraic model logics have been published; see the surveys by Cîrstea et al. [6] and by Kupke and Pattinson [23] for an overview. Both these surveys argue that coalgebras provide the ideal abstract framework to reason about modal logics. We have attempted to use the *predicate-lifting* framework which authors such as Pattinson [31][30], Schröder [41] and Klin [22] develop for automatically creating an expressive modal logic for endofunctors on **Set**, **Set** and **Set**-like categories respectively, but the output did not match our design goals. Our manually designed modal logic instead uses a coalgebraic model of CA *indirectly* to define the semantics, which is incompatible with the standard framework. Most of the existing meta-theory therefore not apply to our logic.

Bisimulations We prove a CA analogue of the Hennessy-Milner (HM) theorem [12]. The original Hennessy-Milner theorem states that two labelled transition systems (LTS) satisfy the same model logic formulas if and only if they are *bisimilar*. This latter notion of bisimilarity is a special case of many coalgebraic definitions of bisimilarity (see Staton [43] for an overview). Our HM theorem does not use the standard definition of bisimilarity, but a novel CA definition of *simulation*. Our definition extends a one-way embedding correspondence (a *simulation* in [40]) with requiring the preservation of a cell's behaviour dynamics. It is unclear if our definition is a special case of the simulations defined by Hughes and Jacobs in [13]. Unlike the original HM theorem, we first prove that one-directional simulation is equivalent to embedding of the logic, and the two-way case follows by applying this in both directions. We also allow simulations and logical equivalence to include a in injection on the underlying signatures (of the coalgebras and logical formulas, respectively).

Path logic of concurrent systems Joyal, Nielsen and Winskel [18] also define a uniform modal logic for paths (analogous to CA traces) for various concurrent systems, such as Labelled Transition Systems and Event Structures, and prove a HM theorem for a generalised definition of simulation. Our work is orthogonal to this, since there seems to be no direct correspondence between CA and the considered concurrent systems.

Temporal Description Logic for CA Delivorias et al. [7] construct a decidable Temporal Description Logic for describing the evolution of states of a CA on $\mathbb{Z}^2$, with semantics given via knowledge bases. Like ours, their language has atomic formulas for states, includes time modalities and an "until" operator, and they give similar examples of expressible properties. The main differences are: (1) we use coalgebraic semantics that apply to any graph (not only to $\mathbb{Z}^2$) and to non-uniform CA, (2) our logic is undecidable, and (3) our language has additional space modalities.

Simulations between CA A CA $\mathcal{B}$ *simulates* a CA $\mathcal{A}$ if the configurations of $\mathcal{A}$ can be injectively embedded into the configurations of $\mathcal{B}$, in such a way that the dynamics are preserved. A CA $\mathcal{B}$ is *intrinsically universal* if it can simulate every CA in a given class. Various authors use different choices of the precise definition of *simulation*. These choices vary in the constraints on the embedding and how they define the preservation of the dynamics. We only give a partial account, and focus on the most general definitions. A historic overview of the study of simulation and universality is given by Ollinger in [29], and §5.2 therein gives a very a general definition of *direct simulation* (given for CA on grids of the form $\mathbb{Z}^d$ for some $d \in \mathbb{N}$, but it applies to all CA). Simulations between CA on Cayley graphs have been studied by Roká [36]. For a discussion on different definitions of intrinsic universality, see [9]. Also non-injective simulations have been studied, for example in [14].

The definition of *simulation* used in this thesis (Definition 8.15) is in some sense more general than the definitions cited above, as applies to many classes of CA (all non-uniform CA, which includes uniform CA on $\mathbb{Z}^d$ and all CA on Cayley graphs), but it is more restricted in what kinds of simulations it permits. In particular, our notation of simulation does imply Ollinger's definition of direct simulation, but not vice versa.

Categorical models of CA The literature provides a few other categorical models of CA, all different from the current model. Capobianco and Uustalu have modelled CA as morphisms between coalgebras of a comonad on the category of uniform spaces [4], as well as morphisms between coalgebras of a *graded* comonad on **Set** [5]. We have attempted to adapt this latter construction to the standard coalgebraic modal logic framework, which unfortunately only yielded a trivial logic. In the present work, CA are modelled as coalgebra objects themselves and not as morphisms between coalgebras.

Widemann and Hauhs [48] have published a bialgebraic model specific to two-dimensional CA. Their focus lies on the theoretical relation between different programming styles for implementing CA.

Non-uniform CA In hindsight we discovered that non-uniform CA have been well-studied: see [1] for an overview.

2 Preliminaries

This section fixes the notations used in this thesis and recites necessary background concepts. Many topics are too large to cover fully, but we try to provide references for further reading where possible. We assume that the reader is familiar with basic category theory. For a general introduction to category theory, see for example one of the books [26], [35] or [25].

2.1 Sets and functions

We fix the following notational conventions for set-theoretical concepts.

For a function $f\colon A \to B$ and a subset $C \subseteq A$, the function $f \restriction C\colon C \to B$ is the domain restriction of f, $f[C] \stackrel{\text{def}}{=} \{f(c) \mid c \in C\}$ is the f-image of C. The preimage of f on C is partitioning of C in sets of elements that f maps to the same point in B:

$$\underset{C}{\mathsf{PreIm}}(f) \stackrel{\text{def}}{=} \{\{c \in C \mid f(c) = b\} \mid b \in f[C]\}.$$

For functions $f\colon A \setminus B \to C$ and $g\colon B \to C$, we write $f \cup g\colon A \to C$ to denote the function

$$x \mapsto \begin{cases} f(x) & x \in A \setminus B \\ g(x) & x \in B. \end{cases}$$

We will sometimes use λ-notation to define a function. As an example of λ-notation, $\lambda n \,.\, n+1$ denotes the map $\mathbb{N} \to \mathbb{N}$ defined as $n \mapsto n+1$. Sometimes we explicitly annotate the domain of variables, e.g., as in $\lambda n \in \mathbb{N} \,.\, n+1$. Functions $A \to B$ that always output the same element $b \in B$ (i.e., those of the form $\lambda a \,.\, b$) will also be denoted as const_b.

The set $\mathbb{1}$ denotes the set with exactly one element. The powerset of a set A is denoted as $\wp(A)$. Set difference is denoted with $\setminus$, defined as:

$$A \setminus B \stackrel{\text{def}}{=} \{a \in A \mid a \notin B\}.$$

Partition orderings Similar to how a complete pizza can be cut into slices or a continent be divided into countries, a set can *partitioned* into mutually disjoined subsets that together form the whole set:

Definition 2.1: Partitioning

A *partitioning* of a set $A \in \mathbf{Set}$ is a set $B \subseteq \wp(A)$ such that every $a \in A$ is included in *exactly one* set in B (so $\bigcup B = A$), and $\emptyset \notin B$. Elements of B are called *partitions*. For every $a \in A$, we denote the partition in B containing a by $|a|_B$.

The *set of all partitionings* of A is denoted by

$$\mathsf{Part}(A) \stackrel{\text{def}}{=} \{B \in \wp(B) \mid \bigcup B = A,\ \emptyset \notin B,\ |n|_B, |m|_B \in B \Rightarrow (|n|_B = |m|_B) \vee (|n|_B \cap |m|_B = \emptyset)\}.$$

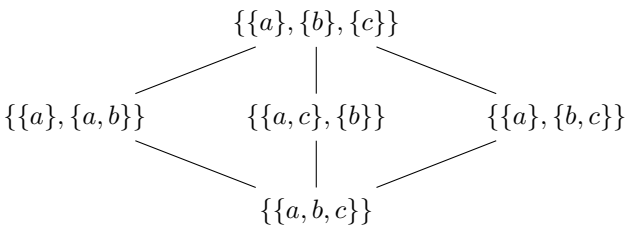
We define the following partial ordering on $\mathsf{Part}(A)$: for partitionings B and C, we have $B \leq C$ if and only if C is further "cut up" partitioning of A than B is: every partition in C can be obtained as a subset of a partition in C. This is equivalent to saying that $a, a' \in |a|_C$ only occurs if $a, a' \in |a|_B$. Formally:

Definition 2.2: Partitionings ordering

The *partitionings ordering* of a set A, denoted as $\mathsf{PartOrd} = (\mathsf{Part}(A), \leq)$, is the poset where $\leq$ is defined for $B, C \in \mathsf{Part}(A)$ as:

$$B \leq C \;\Leftrightarrow\; \bigvee_{a \in A} |a|_C \subseteq |a|_B \,.$$

For example, the Hasse-diagram of $\mathsf{PartOrd}(\{a,b,c\})$ is:



(1)

2.2 Cellular automata and monoids

A cellular automaton (CA) is a computational model whose static structure consists of a directed graph with labelled directed edges whose vertices, called *cells*.

In the dynamics, every cell is assigned a state (often visualised as a colour) from a chosen set of states. Such an assignment of states to cells is called a *configuration*. Every CA comes equipped with (1) a subset of edge labels called a *neighbourhood*, such that every cell has an outgoing edge for every label in the neighbourhood, and (2) a *local rule* that tells how a cell updates it state depending on the states it can see in its local neighbourhood (so the local rule maps an assignment of states to neighbourhood labels to a single state). The local rules define a *global rule* that maps configurations to configurations as follows: the cells synchronously read the states of their neighbours according to the input configuration, feed this to the local rule, and use the output as their state in the output configuration. By iterating the global rule on a given *initial configuration*, one obtains a *trace* of successor configurations. This can be seen as a computation with the initial configuration as input. For general introductions to CA, see for example the books [8] and [37] or the survey [20].

A classical example is Conway's Game of Life (CGOL) [11][9]. This CA uses the regular 2D grid $\mathbb{Z}^2$ as a graph, and as neighbourhood the 3×3 region around a cell (including the central cell itself). It uses only two states: "dead" (white) and "alive" (black). The local rule has an intuitive description:

> *If you are dead, you will become alive the next timestep if you have 3 or 4 alive neighbours. If you are alive, you will stay alive the next timestep if you have 2 or 3 alive neighbours. In all other cases you will be dead the next timestep.*

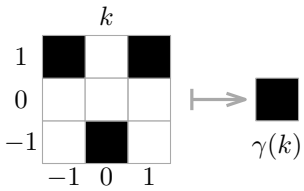
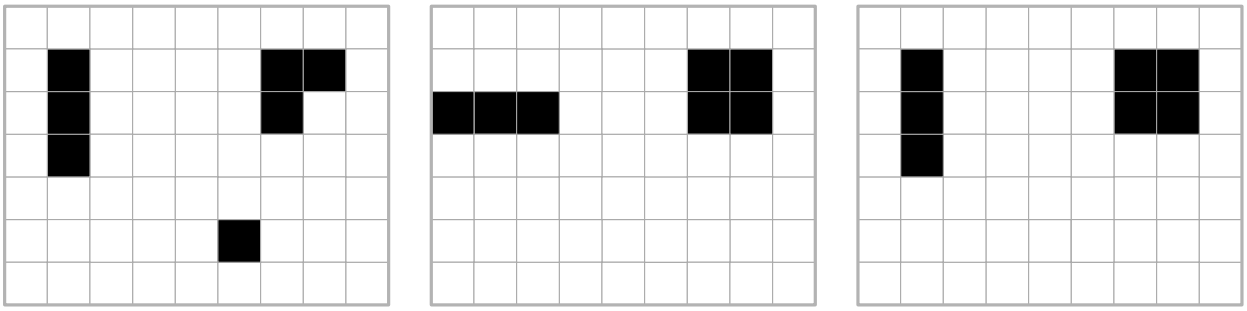


Figure 1: Example of one of CGOL's rules: a dead cell (the central cell, with relative coordinates $(0,0)$) with three alive neighbours will become alive next timestep.



c_0 c_1 c_3

Figure 2: First three configurations in a trace on CGOL. Live cells are visualised as black, dead cells as white. Only parts of the configurations are shown, the other cells are assumed to be white.

To give a precise description of the classical definition of a CA, we fix the convention of representing grids by monoids.

Definition 2.3: Monoid

A monoid $(\mathscr{M}, \bullet, \mathbf{1})$ is a set of elements $\mathscr{M}$ together with an associative binary *multiplication* operation $\bullet\colon \mathscr{M} \times \mathscr{M} \to \mathscr{M}$ and a distinguished *identity element* $\mathbf{1} \in \mathscr{M}$. The multiplication operation satisfies, for all $m, n, \ell \in \mathscr{M}$, the *monoid associativity law*:

$$m \bullet (n \bullet \ell) = (m \bullet n) \bullet \ell$$

and the *monoid unit laws*:

$$\mathbf{1} \bullet m = m = m \bullet \mathbf{1}.$$

Remark 2.4. We typeset monoids in a thin calligraphic font, e.g., $\mathscr{M}$. Often we use the same symbol for both the entire monoid as well as the underlying set of elements. Expressions such as $X^{\mathscr{M}}$, $m \in \mathscr{M}$ and $N \subseteq \mathscr{M}$ *always* refer to the underlying set of elements.

Example 2.5. The natural numbers with addition $(\mathbb{N}, +, 0)$ form a monoid.

Definition 2.6: Freely generated monoid

For any set X, let X^* be the set of finite sequences of elements from X (called *strings*):

$$X^* \stackrel{\text{def}}{=} \{[x_1, x_2, \ldots, x_k] \mid k \in \mathbb{N}, x_1, \ldots, x_k \in X\}.$$

Let $\varepsilon = [\,]$ be the empty string, and let $\bullet_X\colon X^* \times X^* \to X^*$ be string concatenation:

$$[x_1, x_2, \ldots, x_k] \bullet_X [y_1, y_2, \ldots, y_\ell] \stackrel{\text{def}}{=} [x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_\ell]$$

(for all $k, \ell \in \mathbb{N}$ and $x_1, \ldots, x_k, y_1, \ldots, y_\ell \in X$). Then $(X^*, \bullet_X, \varepsilon)$ is a monoid called the *monoid freely generated over* X.

Example 2.7. If we encode $0 = \varepsilon$, $1 = \mathsf{S}$, $2 = \mathsf{SS}$, $3 = \mathsf{SSS}$, etc., and denote string concatenation with $+$, then $\mathbb{N}$ is freely generated over $\{\mathsf{S}\}$.

Definition 2.8: Generated monoid

A *monoid congruence* $\sim$ on a monoid $(\mathscr{M}, \bullet, \mathbf{1})$ is an equivalence relation $\sim \subseteq \mathscr{M} \times \mathscr{M}$ such that for all $m, n, n' \in \mathscr{M}$, if $n \sim n'$ then also $m \bullet n \sim m \bullet n'$ and $n \bullet m \sim n' \bullet m$.

A *monoid generated* over a set X and a relation $R \subseteq X^* \times X^*$ is the monoid $(X^*/{\sim_R}, \bullet_R, |\varepsilon|_{\sim_R})$ where

- $(X^*, \bullet, \varepsilon)$ is the monoid freely generated over X.
- $\sim_R \subseteq X^* \times X^*$ is the monoid congruence

$$\sim_R \stackrel{\text{def}}{=} \bigcap \{\sim \mid R \subseteq \sim,\ \sim \text{ is a monoid congruence on } (X^*, \bullet, \varepsilon)\}.$$

- $|n|_{\sim_R}$ for $n \in X^*$ denotes the equivalence class of n under $\sim_R$.
- The multiplication $\bullet_R$ is:

$$|n|_{\sim_R} \bullet_R |m|_{\sim_R} \stackrel{\text{def}}{=} |n \bullet m|_{\sim_R}.$$

  This is well-defined since $\sim_R$ is a monoid congruence.

Remark 2.9. Relations are often written as equalities. E.g., if $R = \{(AB, C)\}$ then we write $R = \{AB = C\}$.

Example 2.10. Encode $0 = \varepsilon$, $1 = \mathsf{S}$, $2 = \mathsf{SS}$, $3 = \mathsf{SSS}$, etc., and $-1 = \mathsf{P}$, $-2 = \mathsf{PP}$, $-3 = \mathsf{PPP}$, etc. Then the integers $\mathbb{Z}$ are generated over $\{\mathsf{S}, \mathsf{P}\}$ and the relation $R = \{\mathsf{SP} = \varepsilon\}$. Addition is string concatenation modulo $\sim_R$.

A homomorphism between monoids is a map of elements that preserves monoid multiplication.

Definition 2.11: Monoid homomorphism

A *monoid homomorphism* $h\colon (\mathscr{M}, \bullet, \mathbf{1}) \to (\mathscr{M}', \bullet', \mathbf{1}')$ between two monoids $(\mathscr{M}, \bullet, \mathbf{1})$ and $(\mathscr{M}', \bullet', \mathbf{1}')$ is a function $h\colon \mathscr{M} \to \mathscr{M}'$ such that

$$h(\mathbf{1}) = \mathbf{1}' \tag{2}$$

and

$$h(m \bullet n) = h(m) \bullet' h(n)$$

for all $m, n \in \mathscr{M}$.

Definition 2.12: Category of monoids

The category **Mon** has as objects monoids and as morphisms monoid homomorphisms.

We can now state the definition of the static structure of a CA:

Definition 2.13: Cellular automaton (classical)

A *cellular automaton* (CA) $\mathfrak{A}$ is a 4-tuple $\mathfrak{A} = (\mathscr{M}, N, S, \gamma)$, where:

- $(\mathscr{M}, \bullet, \mathbf{1})$ is a monoid that represents the graph;
- $N \subseteq \mathscr{M}$ is the *neighbourhood*. We require that $\mathbf{1} \in N$;
- S is a set of *states*;

- $\gamma\colon S^N \to S$ is a *local rule*.

We interpret elements $x \in \mathscr{M}$ as cells. But an element $m \in \mathscr{M}$ also represents an edge label for the edge between x and $m \bullet x$ (for every $x \in \mathscr{M}$).

Remark 2.14. This is a very general definition of a CA. Many authors limit the allowed monoids to relatively regular grids. For example, only monoids generated over a finite set. Or, even stronger, only those of the form $\mathbb{Z}^d$ with $d \in \mathbb{N}$. The present definition allows monoids that are not grid-like at all, such as binary trees.

Remark 2.15. Not all authors require that $\mathbf{1} \in N$. Ultimately this makes little difference; one can always define γ to ignore the state at $\mathbf{1}$ (i.e., $\gamma(z) = \gamma(z')$ for all $z, z'\colon N \to S$ that only differ on input $\mathbf{1}$).

The static structure of a CA determines the dynamics given by the global rule as follows:

Definition 2.16: Global rule of a CA (classical)

An assignment of states to cells $c\colon \mathscr{M} \to S$ is called a *configuration*. Every CA $\mathfrak{A}$ induces a map $G_{\mathfrak{A}}\colon S^{\mathscr{M}} \to S^{\mathscr{M}}$ between configurations called the *global rule*. For every cell $m \in \mathscr{M}$, it is defined as follows:

$$G_{\mathfrak{A}}(c)(m) \stackrel{\text{def}}{=} \gamma(\lambda n \in N\ .\ c(n \bullet m)). \tag{3}$$

An initial configuration $c_0\colon \mathscr{M} \to S$ induces an sequence of configurations $(c_i)_{i\in\mathbb{N}}$ called the *trace sequence*, which is inductively defined as $c_i \stackrel{\text{def}}{=} G^i_{\mathfrak{A}}(c_0)$.

Example 2.17 (Conway's Game Of Life (CGOL))**.** We can now describe CGOL formally. The Game of Life is the CA

$$\mathsf{CGOL} \stackrel{\text{def}}{=} (\mathbb{Z}^2, N, \{\bullet, \circ\}, \gamma), \tag{4}$$

where $N = \{(i,j) \mid -1 \le i,j \le 1\}$ is the 3×3 rectangle surrounding a cell, and where the state set $\{\bullet, \circ\}$ has only a "dead" ($\circ$) and "alive" ($\bullet$) state. For every input $k\colon N \to \{\bullet, \circ\}$ of the local rule, denote $A(k) \stackrel{\text{def}}{=} \mathsf{card}\{(i,j) \in N \mid k((i,j)) = \bullet\}$. Then the local rule $\gamma\colon \{\bullet, \circ\}^N \to \{\bullet, \circ\}$ can be defined as:

$$\gamma(k) \stackrel{\text{def}}{=} \begin{cases} \circ & A(k) \le 2 \\ \bullet & A(k) = 3 \\ k((0,0)) & A(k) = 4 \\ \circ & A(k) \ge 5 \end{cases} \tag{5}$$

2.3 Category theory

We briefly review concepts from category theory, to start with coalgebras and comonads. For general introduction into coalgebras, see for example [15], [39], [38] or [16]. More background on comonads can be found in textbooks such as [26, Ch6] or [35, Ch5].

Definition 2.18: Coalgebra

A *coalgebra* of an endofunctor $F\colon \mathcal{C} \to \mathcal{C}$ on a category $\mathcal{C}$ an F-coalgebra is a pair $\mathcal{X} = (X, d)$ where $X \in \mathcal{C}$ is an object (referred to as the *carrier*) and $d\colon X \to FX$ a morphism in $\mathcal{C}$ (referred to as the *dynamics*, or (if clear from context) as the *coalgebra*).

A *coalgebra homomorphism* between two F-coalgebras (X, d) and (Y, e) is a morphism $h\colon X \to Y$ in $\mathcal{C}$

such that commutes:

$$\begin{array}{ccc} X & \xrightarrow{h} & Y \\ {\scriptstyle d}\downarrow & & \downarrow{\scriptstyle e} \\ FX & \xrightarrow[Fh]{} & FY \end{array} \tag{6}$$

This diagram will be called the *coalgebra homomorphism law*.

The coalgebras of F together with the coalgebra homomorphisms form a category $\mathsf{Coalg}(F)$.

We typeset coalgebras in a thick calligraphic font, e.g., $\mathcal{X}$. If a coalgebra name contains multiple symbols we use overarrows instead, e.g., $\overrightarrow{\mathsf{CGOL}}$.

If the endofunctor F is a *comonad*, then one can construct the *coEilenberg-Moore* category $\mathsf{coEM}(F)$ of comonad-coalgebras, which is a subcategory of $\mathsf{Coalg}(F)$.

Definition 2.19: Comonad

A *comonad* is an endofunctor $F\colon \mathcal{C} \to \mathcal{C}$ equipped with two natural transformation $\varepsilon_X\colon FX \to X$ (called the *counit*) and $\delta_X\colon FX \to FFX$ (called the *comultiplication*) such that the following diagrams commute for all $X \in \mathcal{C}$:

$$\begin{array}{ccc} FX & \xrightarrow{\delta_X} & FFX \\ {\scriptstyle \delta_X}\downarrow & & \downarrow{\scriptstyle F\delta_X} \\ FFX & \xrightarrow[\delta_{FX}]{} & FFFX \end{array} \tag{7}$$

$$\begin{array}{ccccc} & & FX & & \\ & \swarrow\!\!\!\swarrow & \downarrow{\scriptstyle \delta_X} & \searrow\!\!\!\searrow & \\ FX & \xleftarrow[\delta_{FX}]{} & FFX & \xrightarrow[F\delta_X]{} & FC \end{array} \tag{8}$$

which will be referred to as the *comonad coassociativity law* and *comonad counit law* respectively.

Definition 2.20: coEilenberg-Moore coalgebras

The category $\mathsf{coEM}(F)$ of comonad $F\colon \mathcal{C} \to \mathcal{C}$ is the subcategory of $\mathsf{Coalg}(F)$ containing only the objects (X, d) such that the following diagrams commute:

$$\begin{array}{ccc} X & \xrightarrow{d} & FX \\ {\scriptstyle d}\downarrow & & \downarrow{\scriptstyle Fd} \\ FX & \xrightarrow[\delta_X]{} & FFX \end{array} \tag{9}$$

$$\begin{array}{ccc} X & \xrightarrow{d} & FX \\ & \searrow\!\!\!\searrow & \downarrow{\scriptstyle \varepsilon_X} \\ & & X \end{array} \tag{10}$$

These diagrams will be referred to as the *comonad coalgebra coassociativity law* and the *comonad coalgebra counit law*, respectively.

For coalgebras on endofunctors of **Set** these exists a natural notion of similarity between elements of their carriers, known as *behavioural equivalence*. See Staton [43] for an overview of related definitions generalised beyond **Set**.

Definition 2.21: Behavioural equivalence

Let $\mathcal{X} = (X, d)$ and $\mathcal{Y} = (Y, e)$ be F-coalgebras. Let $x \in X$ and $y \in Y$. Then x is *behaviourally equivalent* to y, denoted $x \overset{\mathsf{BE}}{\approx} y$, if there exist a coalgebra (Z, f) and coalgebra homomorphism $h\colon (X, d) \to (Z, f)$ and $k\colon (Y, e) \to (Z, f)$ such that $h(x) = k(y)$.

Remark 2.22. For any coalgebra $\mathcal{Y}$ there always exists an identity endohomomorphism $\mathsf{Id}_{\mathcal{Y}}\colon \mathcal{Y} \to \mathcal{Y}$. As a result, for any coalgebra homomorphism $h\colon \mathcal{X} \to \mathcal{Y}$ it automatically holds that $x \overset{\mathsf{BE}}{\approx} h(x)$ for all $x \in X$ (this uses $k := \mathsf{Id}_{\mathcal{Y}}$).

In this thesis we use functors defined in terms of dependent sums in **Set**, for which we use the following definition:

Definition 2.23: Dependent sums of sets

Given a set A and a A-indexed family of sets B the dependent sum $\sum_{a \in A} B(a)$ is defined as the set of tuples (a, b) such that $b \in B(a)$ (the set $B(a)$ from which the second element is drawn depends on the first element a).

Remark 2.24. The set $\sum_a B(A)$ is the coproduct in **Set** of the diagram

$$D\colon \Delta_A \to \mathbf{Set}$$
$$D\colon \Delta_A \mapsto \{a\} \times B(a)$$

where Δ_A is the small discrete category with elements of A as objects and with only identity morphisms.

2.4 Modal logic

We review only the basic definitions and relevant results of modal logic. We refer the reader to the textbook by Blackburn, De Rijke and Venema [2] for a comprehensive introduction.

A Kripke modal logic is an extension of propositional (Boolean) logic with unary operators $\diamond_i$ known as *modal operators*.

Definition 2.25: Grammar of modal logic

A *signature* for modal logics is a pair (τ, Φ) where τ is a set of unary operators $(\diamond_i)_{i \in \tau}$ and $\Phi = \{p, q, r, \dots\}$ a set of propositional letters (equivalent to Boolean variables).

The set of well-formed formulas $\mathsf{Form}(\tau, \Phi)$ on a signature (τ, Φ) is:

$$\mathsf{Form}(\tau, \Phi) ::= \quad \neg\psi \mid \psi_1 \vee \psi_2 \mid \bot \mid p \mid \diamond_i \psi$$

where $p \in \Phi$ and $i \in \tau$. We abbreviate $\neg \diamond_i \neg\psi$ as $\Box_i \psi$.

One can generalise the diamonds to be n-ary operators for any $n \in \mathbb{N}$, but this generalisation is not relevant for the present work.

Modal formulas are interpreted at points in relational structures, also known as *Kripke models*.

Definition 2.26: Kripke frame and models

Let (τ, Φ) be a signature for a Kripke modal logic.

A *Kripke frame* $\mathbb{F} = (W, R_i)_{i \in \tau}$ consists of a set W *points* (also called *worlds*), and a τ-indexed set of

binary relations $(R_i \subseteq W \times W)_{i\in\tau}$.

A *Kripke model* $(\mathbb{F}, V)$ consists of a Kripke frame $\mathbb{F} = (W, R_i)_{i\in\tau}$ together with a *valuation* $V\colon W \to \wp(\Phi)$.

A formula $\psi \in \mathsf{Form}(\tau, \Phi)$ is either true (*satisfied*) or false (*unsatisfied*) when interpreted at a specific point $w \in W$ of a model $((W, R_i)_{i\in\tau}, V)$. Propositional letters are true at w when they occur in $V(w)$. However, the truth of ψ at w does not only depend on $V(w)$: the modal operators $\diamond_i\psi'$ allow to evaluate the truth a subformula ψ' at a R_i-neighbour v (i.e., $(w, v) \in R_i$). In particular, $\diamond_i\psi'$ is true at w as soon as there exist *any* R_i neighbour of w where ψ' is true. Dually, $\Box_i\psi'$ is true at w if ψ' is true at *all* R_i-neighbours of w.

Definition 2.27: Kripke semantics modal logic

Given a Kripke model $M = (\mathbb{F} = (W, R_i)_{i\in\tau}, V)$ defined on a signature (τ, Φ), we define for each $w \in W$ and formula $\psi \in \mathsf{Form}(\tau, \Phi)$ the proposition $M, w \vDash \psi$ to either be true or false (synonyms: ϕ is either *satisfied* or *unsatisfied* at w in M). This truth is defined by structural induction on ψ as follows:

$$
\begin{aligned}
&M, w \vDash \neg\psi && \text{iff it is false that } M, w \vDash \psi\\
&M, w \vDash \psi_1 \vee \psi_2 && \text{iff } (M, w \vDash \psi_1 \text{ or } M, w \vDash \psi_2)\\
&M, w \vDash \bot && \text{is always false}\\
&M, w \vDash p && \text{iff } p \in V(w)\\
&M, w \vDash \diamond_i\psi && \text{iff there exists a } v \in W \text{ such that } (w, v) \in R_i \text{ and } M, v \vDash \psi.
\end{aligned}
$$

A formula ψ is:

- *globally satisfied in a model* M if $M, w \vDash \psi$ for all $w \in W$.
- *valid in a frame* $\mathbb{F}$ if $(\mathbb{F}, V), w \vDash \psi$ for all $V\colon W \to \wp(\Phi)$ and $w \in W$.
- *valid at a point* $w \in W$ if $(\mathbb{F}, V), w \vDash \psi$ for all $V\colon W \to \wp(\Phi)$.

Remark 2.28. The semantics of formulas without modal operators are the same as in propositional logic, which justifies the following familiar notational shorthands (for all $\psi_1, \psi_2 \in \mathsf{Form}(\tau, \Phi)$):

$$
\begin{aligned}
\psi_1 \wedge \psi_2 &\stackrel{\text{def}}{=} \neg((\neg\psi_1) \vee (\neg\psi_2))\\
\psi_1 \to \psi_2 &\stackrel{\text{def}}{=} (\neg\psi_1) \vee \psi_2\\
\psi_1 \leftrightarrow \psi_2 &\stackrel{\text{def}}{=} (\psi_1 \to \psi_2) \wedge (\psi_2 \to \psi_1)\\
\top &\stackrel{\text{def}}{=} \neg\bot.
\end{aligned}
$$

An important result of modal logic is the Hennessy-Milner theorem [12]. Intuitively, the theorem asserts that two points $w, v \in W$ of a model satisfy the same set of formulas if and only if they have a similar successor structure (by recursively following outgoing relations R_i) in the model. The latter property is called *bisimilarity*, and the precise definition is as follows:

Definition 2.29: Bisimulation and logical equivalence

Let (τ, Φ) be a modal logic signature. Let $M = (W, (R_i)_{i\in\tau}, V)$ and $M = (W', (R'_i)_{i\in\tau}, V')$ be Kripke models. A bisimulation $Z\colon M \leftrightarrows M'$ is a binary relation $Z \subseteq W \times W'$ that satisfies the following three conditions:

1. If uZu' then $V(u) = V'(u')$ (for all $u \in W$ and $u' \in W'$).
2. If uR_iv and uZu' then there exists a $v' \in W$ such that $u'R_iv'$ (for all $u, v \in W$, $u' \in W'$ and $i \in \tau$).
3. If $u'R'_iv'$ and uZu' then there exists a $v \in W$ such that uR_iv (for all $u \in W$, $u', v' \in W'$ and $i \in \tau$).

Points $w \in W$ and $w' \in W$ are *bisimilar*, denoted $w \leftrightarroweq w'$, if there exists a bisimulation $Z \colon M \leftrightarroweq M'$ such that wZw'. The points are *logically equivalent*, denoted $w \leftrightsquigarrow w'$, if for all $\psi \in \mathsf{Form}(\tau, \Phi)$ it holds that

$$(M, w \vDash \psi) \quad \text{iff} \quad (M', w' \vDash \psi).$$

Definition 2.30: Image finite frame

A Kripke frame $\mathbb{F} = (W, (R_i)_{i\in\tau})$ is *image-finite* if for all relations R_i and all points $w \in W$ the set

$$R_i(w) \stackrel{\text{def}}{=} \{v \in W \mid wR_iv\}$$

is finite.

Theorem 2.31: Hennessy and Milner, 1984, [12]

Let (τ, Φ) be a Kripke modal logic signature, $M = (W, (R_i)_{i\in\tau}, V)$ and $M = (W', (R'_i)_{i\in\tau}, V')$ be image-finite Kripke models and $w \in W$, $w' \in W'$ be points. Then

$$w \leftrightarroweq w' \quad \text{iff} \quad w \leftrightsquigarrow w'.$$

3 Uniform cellular automata

In this section we exhibit the class of CA as a subcategory $\mathbf{CCA}^+_{\mathsf{U}}$ of the coalgebras of an endofunctor $C_{\mathsf{U}} \colon \mathbf{Set} \to \mathbf{Set}$. The main result, Theorem 3.17, states that there is an essentially surjective functor from classical CA c.f. Definition 2.13 to $\mathbf{CCA}^+_{\mathsf{U}}$. This coalgebraic formalisation will be refined in section 4, and thereafter feature in the semantics of the modal logic described in Sections 6 and 7.

3.1 Graph as coalgebras

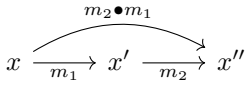
We first sketch the intuition of our coalgebraic model of CA. Unlike the classical definition of a CA (Definition 2.13) and Capobianco's and Uustalu's [5][4] model, we will distinguish between cells and paths between cells. To describe how cells are interconnected, we use the same *cowriter comonad* as used by Capobianco and Uustalu (in a limited form described for CA before by Piponi [34]). To model paths, we use a monoid $(\mathscr{M}, \bullet, \mathbf{1})$, and interpret the monoid operation $m \bullet n$ as path composition (so traversing $m \bullet n$ is equivalent to first traversing n and m thereafter).

The comonad works as follows: a coalgebra (X, χ) consists of a carrier X, the set of cells, and a mapping $\chi \colon X \to X^{\mathscr{M}}$. Unlike Capobianco and Uustalu, we interpret X as the set of cells instead of the set of states. We take the convention to write χ as an infix operator $\circledast$. For every path $m \in \mathscr{M}$ and cell $x \in X$, we compute the cell one arrives at by traversing the path m from the initial cell x as $m \circledast x$.

We will show that the comonad structure ensures that the graph structure is coherent. First of all, traversing the empty path $\mathbf{1} \in \mathscr{M}$ should not lead to a different cell. In particular, we require that the following identity holds for all $x \in X$:

$$\mathbf{1} \between x = x. \tag{11}$$

We further require that $\between$ respects path composition. Suppose that m_1 is a path from x to x', and that m_2 is a path from x' to x''. Then in the $\bullet$-is-path-composition interpretation it should hold that $m_2 \bullet m_1$ is a path from x to x''.



This means that we require (for all $m_1, m_2 \in \mathscr{M}$ and all $x \in X$):

$$m_2 \between (m_1 \between x) = (m_2 \bullet m_1) \between x. \tag{12}$$

(From now on, we will usually omit parenthesis in expressions of the form $m_2 \between m_1 \between x$).

Requiring χ to adhere to conditions Eq. (11) and (12) turns out to be equivalent to requiring (X, χ) to be a coalgebra in the coEilenberg-Moore category (Definition 2.20) of the following cowriter comonad:

Definition 3.1: Cowriter functor

The *cowriter functor* $P = P\colon \mathbf{Set} \to \mathbf{Set}$ is the functor whose action on objects $X \in \mathbf{Set}$ is

$$P\colon X \mapsto X^{\mathscr{M}} \tag{13}$$

and whose action on functions $f\colon X \to Y$ is:

$$Pf \stackrel{\text{def}}{=} \lambda k \,.\, f \circ k\colon X^{\mathscr{M}} \to Y^{\mathscr{M}} \tag{14}$$

for all $k\colon \mathscr{M} \to X$.

To make P a comonad, we define the associated natural transformations:

Definition 3.2: Comonad structure of the cowriter functor

The counit ε and comulitplication δ of P are:

$$\begin{aligned} \varepsilon\colon P &\Rightarrow \mathsf{Id} & \qquad \delta\colon P &\Rightarrow PP \\ \varepsilon_X\colon X^{\mathscr{M}} &\to X & \qquad \delta_X\colon X^{\mathscr{M}} &\to (X^{\mathscr{M}})^{\mathscr{M}} \\ \varepsilon_X(k) &\stackrel{\text{def}}{=} k(\mathbf{1}) & \qquad \delta_X(k) &\stackrel{\text{def}}{=} \lambda m \,.\, \lambda n \,.\, k(n \bullet m) \end{aligned} \tag{15}$$

for all $k\colon \mathscr{M} \to X$ and where $m, n \in \mathscr{M}$.

Remark 3.3. Alternatively, we could also define comultiplication δ as

$$\delta_X\colon k \mapsto \lambda m \,.\, \lambda n \,.\, k(m \bullet n). \tag{16}$$

In this case, we should interpret of $m \bullet n$ as the path obtained from first traversing m and thereafter n (for $n, m \in \mathscr{M}$), and it would be more convenient to write $x \between m \between n$ instead of $m \between n \between x$. We do not follow this alternative approach, but it showcases that the interpretation of the order in path composition is arbitrary.

Before we prove that this is indeed a comonad, we check that the comonad coassociativity and counit laws, (9) and (10) respectively, indeed correspond to Eq. (12) and (11) respectively. The first law (9) expresses

that the following diagram commutes:

$$\begin{array}{ccc} X & \xrightarrow{\chi} & X^{\mathscr{M}} \\ {\scriptstyle\chi}\downarrow & & \downarrow{\scriptstyle P\chi} \\ X^{\mathscr{M}} & \xrightarrow[\delta_X]{} & (X^{\mathscr{M}})^{\mathscr{M}} \end{array}$$

On inputs $x \in X$ and $m, n \in \mathscr{M}$, the upper path outputs:

$$\begin{aligned} (P\chi \circ \chi)(x,m,n) &= (P\chi(\chi(x)))(m,n) \\ &= (\chi \circ (\chi(x)))(m,n) && \text{// Definition 3.1.} \\ &= (\chi(m \circledast x))(n) \\ &= n \circledast (m \circledast x). \end{aligned}$$

The lower path outputs:

$$\begin{aligned} (\delta_X \circ \chi)(x,m,n) &= (\delta_X(\chi x))(m,n) \\ &= (\lambda m' \,.\, \lambda n' \,.\, (n' \bullet m') \circledast x)(m,n) && \text{// Definition 3.2.} \\ &= (n \bullet m) \circledast x. \end{aligned}$$

Equality of these paths is indeed exactly requirement (12).

The second law (10) states that the following diagram commutes:

$$\begin{array}{ccc} X & \xrightarrow{\chi} & X^{\mathscr{M}} \\ & \searrow & \downarrow{\scriptstyle \varepsilon_X} \\ & & X \end{array}$$

Then, for every $x \in X$, we have:

$$x = (\varepsilon_X \circ \chi)(x) = \varepsilon_X(\chi(x)) = \chi(x)(\mathbf{1}) = \mathbf{1} \circledast x,$$

as required by (11).

We now show that Definition 3.2 indeed a comonad structure on the cowriter functor P. This result is not new, but we present the details for the reader's convenience:

Lemma 3.4

(P, ε, δ) is a comonad.

Proof. Referring to Definition 3.2, we need to show that $\varepsilon \colon P \Rightarrow \mathsf{Id}$ and $\delta \colon P \Rightarrow PP$ are indeed natural in their arguments, and that the comonad coassociativity law (7) and comonad counit law (8) are satisfied.

As for the naturality of ε, let $f \colon X \to Y$, and it is to show that commutes:

$$\begin{array}{ccc} X^{\mathscr{M}} & \xrightarrow{Pf} & Y^{\mathscr{M}} \\ {\scriptstyle\varepsilon_X}\downarrow & & \downarrow{\scriptstyle\varepsilon_Y} \\ X & \xrightarrow[f]{} & Y \end{array}$$

Indeed, for any $k \colon \mathscr{M} \to X$ we observe that

$$(f \circ \varepsilon_X)(k) = f(k(\mathbf{1})) = (f \circ k)(\mathbf{1}) = (\varepsilon_X \circ Pf)(k).$$

The naturality condition for δ for a function $f\colon X \to Y$ is

$$\begin{array}{ccc} X^{\mathscr{M}} & \xrightarrow{Pf} & Y^{\mathscr{M}} \\ {\scriptstyle \delta_X}\downarrow & & \downarrow{\scriptstyle \delta_Y} \\ (X^{\mathscr{M}})^{\mathscr{M}} & \xrightarrow[PPf]{} & (Y^{\mathscr{M}})^{\mathscr{M}} \end{array}$$

We can indeed calculate that for any input $k\colon \mathscr{M} \to X$ we have:

$$\begin{aligned} (PPf) \circ \delta_X(k) &= PPf(\lambda m\,.\,\lambda n\,.\,k(n \bullet m)) && \text{// Definition } \delta \text{ (Definition 3.2).} \\ &= \lambda m\,.\,Pf \circ \lambda n\,.\,k(n \bullet m) && \text{// Definition outermost } P \text{ (Definition 3.1).} \\ &= \lambda m\,.\,\lambda n\,.\,(f \circ k)(n \bullet m) && \text{// Definition } P \text{ (Definition 3.1).} \\ &= (\delta_Y \circ (Pf))(k). && \text{// Definition } \delta \text{ (Definition 3.2).} \end{aligned}$$

The comonad coassociativity law (7) requires that commutes:

$$\begin{array}{ccc} X^{\mathscr{M}} & \xrightarrow{\delta_X} & (X^{\mathscr{M}})^{\mathscr{M}} \\ {\scriptstyle \delta_X}\downarrow & & \downarrow{\scriptstyle P\delta_X} \\ (X^{\mathscr{M}})^{\mathscr{M}} & \xrightarrow[\delta_{X^{\mathscr{M}}}]{} & ((X^{\mathscr{M}})^{\mathscr{M}})^{\mathscr{M}} \end{array} \tag{17}$$

For an input $k\colon \mathscr{M} \to X$, the upper path evaluates to:

$$\begin{aligned} (\delta_{X^{\mathscr{M}}} \circ \delta_X)(k) &= \lambda u\,.\,\lambda v\,.\,\delta_X(k)(v \bullet u) && \text{// Definition } \delta \text{ (Definition 3.2).} \\ &= \lambda u\,.\,\lambda v\,.\,\lambda m\,.\,\lambda n\,.\,k(n \bullet m)(v \bullet u) && \text{// Definition } \delta \text{ (Definition 3.2).} \\ &= \lambda u\,.\,\lambda v\,.\,\lambda n\,.\,k(n \bullet v \bullet u). \end{aligned}$$

This indeed agrees with the lower path:

$$\begin{aligned} (P\delta_X \circ \delta_X)(k) &= P\delta_X\,[\lambda u\,.\,\lambda w\,.\,k(w \bullet u)] \\ &= \lambda u\,.\,\delta_X(\lambda u\,.\,\lambda w\,.\,k(w \bullet u)) && \text{// Definition } P \text{ (Definition 3.1).} \\ &= \lambda u\,.\,\lambda v\,.\,\lambda n\,.\,(\lambda u\,.\,\lambda w\,.\,k(w \bullet u))(n \bullet v) && \text{// Definition } \delta \text{ (Definition 3.2).} \\ &= \lambda u\,.\,\lambda v\,.\,\lambda n\,.\,k(n \bullet v \bullet u), \end{aligned}$$

which shows that both paths in the diagram indeed agree.

Finally, we need to show that the comonad counit law (8) holds:

$$\begin{array}{ccccc} & & X^{\mathscr{M}} & & \\ & \swarrow & \downarrow{\scriptstyle \delta_X} & \searrow & \\ X^{\mathscr{M}} & \xleftarrow[\varepsilon_{(X^{\mathscr{M}})}]{} & (X^{\mathscr{M}})^{\mathscr{M}} & \xrightarrow[P\varepsilon_X]{} & X^{\mathscr{M}} \end{array} \tag{18}$$

To this end, we show for both inner triangles separately that they commute on any input $k\colon \mathscr{M} \to X$. The following computation shows that the right triangle commutes:

$$\begin{aligned} P\varepsilon_X(\delta_X(k)) &= P\varepsilon_X(\lambda m\,.\,\lambda n\,.\,k(n \bullet m)) \\ &= \lambda m\,.\,\varepsilon_X(\lambda n\,.\,k(n \bullet m)) && \text{// Definition } P \text{ (Definition 3.1).} \\ &= \lambda m\,.\,k(\mathbf{1} \bullet m) && \text{// Definition } \varepsilon \text{ (Definition 3.2).} \\ &= k. \end{aligned}$$

Commutativity of the left triangle follows by a similar computation, using the definitions of δ and ε (Definition 3.2):

$$\varepsilon_{(X^{\mathscr{M}})}(\delta_X(k)) = \varepsilon_{(X^{\mathscr{M}})}(\lambda m\,.\,\lambda n\,.\,k(n \bullet m)) = \lambda n\,.\,k(n \bullet \mathbf{1}) = k.$$

□

3.2 Extending graphs with local rules

The coalgebras of the paths functor P are cells with a graph structure. These structured need to be complemented with local rules before they become CA, and we will build the local rules directly into the coalgebra type functor. To this end, we will modify P into a "uniform CA" functor C_{U} with an additional component for specifying local rules. We will use C_{U} to construct a category of coalgebras that correspond to uniform CA, i.e., CA where all cells use the same local rule, as in Definition 2.13 (the underscript u stands for *uniform*). Section 4 will construct an endofunctor C_{G} and a category of *non-uniform* coalgebras in which cells are allowed to have different local rules.

In the following, we assume an additional parameter next to $\mathscr{M}$: a state set $S \in \mathbf{Set}$.

Definition 3.5

The functor $C_{\mathsf{U}} \colon \mathbf{Set} \to \mathbf{Set}$ is given by the following action on every set $X \in \mathbf{Set}$:

$$C_{\mathsf{U}}(X) \stackrel{\text{def}}{=} X^{\mathscr{M}} \times S^{S^{\mathscr{M}}}$$

and the action on functions $f \colon X \to Y$ is:

$$C_{\mathsf{U}} f \stackrel{\text{def}}{=} \lambda k\,.\,(f \circ k) \times \mathsf{Id}$$

where $k \colon \mathscr{M} \to X$ and $(f \circ k) \colon \mathscr{M} \to Y$.

Let (X, d) be a C_{U}-coalgebra and $x \in X$. Then $d(x) \in X^{\mathscr{M}} \times S^{S^{\mathscr{M}}}$ has two components $\pi_1(d(x))$ and $\pi_2(d(x))$, where π_i denotes the projection to the i^{th} component. All our applications will use these components separately, and we use the following notational convention to ease this: if we define $\chi \stackrel{\text{def}}{=} \pi_1 \circ d$ and $\gamma \stackrel{\text{def}}{=} \pi_2 d$, then we will write $\langle \chi, \gamma \rangle$ for d.

Observe that first component $\chi \colon X \to X^{\mathscr{M}}$ has the same type as a cowriter coalgebra as discussed in Section 3.1. The second component has type

$$\gamma \colon X \to S^{S^{\mathscr{M}}}. \tag{19}$$

Thus γ assigns to every cell $x \in X$ a local rule $\gamma_x \colon S^{\mathscr{M}} \to S$, with neighbourhood $\mathscr{M}$. This is a more general collection of local rules than the local rules in a classical CA according to Definition 2.13: (1) it is possible that $\gamma_x \neg \gamma_{x'}$ for $x, x' \in X$, and (2) the neighbourhood is the entire monoid $\mathscr{M}$, while a CA may have a smaller neighbourhood $N \subseteq \mathscr{M}$.

Example 3.6 (A coalgebraic model of Conway's Game of Life)**.** It is straightforward to describe CGOL (Example 2.17) as a C_{U}-coalgebra $\overrightarrow{\mathsf{CGOL}} \stackrel{\text{def}}{=} (\mathbb{Z}^2, \langle +, \gamma \rangle)$ where $(\mathbb{Z}^2, +, (0,0))$ is the underlying monoid and the used cowriter comonad coalgebra "+" is component-wise addition: $(i,j) \oplus (k,\ell) \stackrel{\text{def}}{=} (i+k, j+\ell)$. The neighbourhood N and local rule γ are exactly as in the classical definition of Example 2.17.

3.3 Uniform cellular automata as coalgebras

We will now construct the category of C_{U}-coalgebras that corresponds to classical CA (Definition 2.13). Different CA may have different underlying monoids $\mathscr{M}$, neighbourhoods $N \subseteq \mathscr{M}$ and state sets S. Therefore

we will need to combine coalgebras of C_{U} for different choices of the parameters $\mathscr{M}$ and S. To avoid ambiguity, we write $C_{\mathsf{U}}^{(\mathscr{M},S)}$ when using C_{U} with parameters $\mathscr{M}$ and S.

We will call the parameters of a CA a *signature*:

Definition 3.7: Signature

A *signature* is a triplet $(\mathscr{M}, N, S)$ where $(\mathscr{M}, \bullet, \mathbf{1})$ is a monoid, $N \subseteq \mathscr{M}$ with $\mathbf{1} \in \mathscr{M}$ a neighbourhood, and $S \in \mathbf{Set}$ a state set.

After an auxiliary definition (of *rooted coalgebras*), we will specify the subcategory $\mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$ of $\mathsf{Coalg}(C_{\mathsf{U}}^{(\mathscr{M},S)})$ of coalgebras that correspond to CA defined on this signature $(\mathscr{M}, N, S)$. Thereafter we will combine these subcategories (over all choices of signatures) in one category $\mathbf{CCA}_{\mathsf{U}}^{+}$, and show that the latter exactly corresponds to the collection of classical CA conform Definition 2.13.

For any $M \subseteq \mathscr{M}$, a coalgebra $(X, \langle \chi, \gamma \rangle)$ and an $x \in X$, we will denote the M-image of $\chi(x)$ as:

$$M \circledast x \stackrel{\text{def}}{=} \{m \circledast r \mid m \in M\}. \tag{20}$$

We can now define a *rooted* coalgebra:

Definition 3.8: Rooted coalgebra

A coalgebra $\mathcal{X} = (X, \langle \chi, \gamma \rangle)$ is *rooted* in a distinguished root $r \in X$ if $\mathscr{M} \circledast r = X$.

Remark 3.9. The choice of r may not be unique, and not every coalgebra may have a root.

Definition 3.10

Let $(\mathscr{M}, N, S)$ be a signature, then $\mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$ is the subcategory of $\mathsf{Coalg}(C_{\mathsf{U}}^{(\mathscr{M},S)})$ whose objects $\mathcal{X} = (X, \langle \chi, \gamma \rangle)$ satisfy the following constraints:

1. There exists an *essential local rule* $\mathring{\gamma} \colon S^N \to S$ such that $\gamma_x(k) = \mathring{\gamma}(k \restriction N)$ for all $x \in X$ and all $k \colon \mathscr{M} \to S$.
2. $\mathcal{X}$ is rooted in a chosen root $r \in X$.
3. $(-) \circledast r \colon \mathscr{M} \to X$ is an injection.
4. (X, χ) is a cowriter comonad coalgebra (of the functor P, Definition 3.1 and 3.2).

We usually make the essential local rule $\mathring{\gamma}$ and the root $r \in X$ explicit, in which case we denote a coalgebra in $\mathbf{CCA}_{\mathsf{U}}$ as $\mathcal{X} = (X, \langle \chi, \gamma \rangle, \mathring{\gamma}, r)$.

The morphisms in $\mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$ are root-preserving coalgebra homomorphisms.

Remark 3.11. Item 1 ensures that the CA has exactly one local rule for all cell (uniformity), and that this rule is also actually local (its output only depends on the states in an N-neighbourhood around x).

Example 3.12. One-dimensional CA are CA on the monoid $(\mathbb{Z}, +, 0)$ with as cells $\mathbb{Z}$ and as cowriter comonad coalgebra the usual addition. Any cell $n \in \mathbb{Z}$ can be chosen as the root: every number $m \in \mathbb{N}$ can be uniquely written as $m = (m - n) + n$, thus $(-) + n$ has $\mathbb{Z}$ as image, and is injective.

Now we combine the $\mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$ over all choices of signature into one category $\mathbf{CCA}_{\mathsf{U}}^{+}$. A technical detail required to prove the correspondence between classical CA and $\mathbf{CCA}_{\mathsf{U}}^{+}$ (in Theorem 3.17) is that we need to

introduce additional arrows aside from existing the coalgebra homomorphisms. Intuitively, these extra arrows are isomorphisms between coalgebras that are identical up to a renaming of the elements of the monoid they are defined on.

Definition 3.13

The category $\mathbf{CCA}_{\mathsf{U}}^{+}$ is the coproduct of the categories $\mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$ over all signatures $(\mathscr{M},N,S)$, extended with the following arrows: for every pair of signatures $(\mathscr{M},N,S)$ and $(\mathscr{M}',N',S)$ with the same set of states, and for every monoid isomorphism $\phi\colon \mathscr{M} \xrightarrow{\sim} \mathscr{M}'$, and for all pairs of coalgebras

$$\mathcal{X} = (X, \langle \chi, \gamma \rangle, \mathring{\gamma}, r_X) \in \mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$$
$$\mathcal{Y} = (Y, \langle \xi, \delta \rangle, \mathring{\delta}, r_Y) \in \mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M}',N',S)},$$

if

1. $X = Y$ and $r_X = r_Y$,
2. $m \mathbin{⦶} x = \phi(m) \mathbin{⦷} x$ for all $x \in X$ and $m \in \mathscr{M}$,
3. and $\mathring{\delta} = \lambda k \,.\, \mathring{\gamma}(k \circ \phi)$,

then there is an arrow $\phi_{\mathcal{X},\mathcal{Y}} \in \mathbf{CCA}_{\mathsf{U}}^{+}(\mathcal{X},\mathcal{Y})$ whose action on cells is the identity map. The inverse of this arrow is $(\phi^{-1})_{\mathcal{Y},\mathcal{X}} \in \mathbf{CCA}_{\mathsf{U}}^{+}(\mathcal{Y},\mathcal{X})$.

Remark 3.14. The isomorphism $\phi_{\mathcal{X},\mathcal{Y}} \in \mathbf{CCA}_{\mathsf{U}}^{+}(\mathcal{X} \to \mathcal{Y})$ witnesses that the following diagram commutes:

$$\begin{array}{ccc} X & \xlongequal{\quad \mathsf{Id} \quad} & Y \\ {\scriptstyle \langle \gamma, \chi \rangle}\downarrow & & \downarrow{\scriptstyle \langle \xi, \delta \rangle} \\ X^{\mathscr{M}} \times S^{S^{\mathscr{M}}} & \xrightarrow[\mathsf{Id}']{} & Y^{\mathscr{M}'} \times S^{S^{\mathscr{M}'}} \end{array} \tag{21}$$

where

$$\mathsf{Id}' \stackrel{\text{def}}{=} (\lambda f \,.\, \mathsf{Id} \circ (f \circ \phi^{-1})) \times (\lambda g \,.\, \lambda k \,.\, g(k \circ \phi)).$$

This is a condition similar to that of a coalgebra homomorphism (compare with diagram (6)), except that it has been extended with the monoid isomorphism ϕ to ensure type consistency. In the special case where $\phi = \mathsf{Id}_{\mathscr{M}}$, diagram 21 simplifies to the coalgebra homomorphism law on $\mathsf{Id}_X\colon \mathcal{X} \to \mathcal{X}$, since

$$(\lambda f \,.\, \mathsf{Id}_X \circ (f \circ \mathsf{Id}_{\mathscr{M}}^{-1})) \times (\lambda g \,.\, \lambda k \,.\, g(k \circ \mathsf{Id}_{\mathscr{M}})) = (\mathsf{Id}_X \circ f) \times \mathsf{Id}_{(S^{S^{\mathscr{M}}})} \stackrel{\text{def}}{=} C_{\mathsf{U}}(\mathsf{Id}_X).$$

To prove that $\mathbf{CCA}_{\mathsf{U}}^{+}$ corresponds to the classical CA, we view the classical CA as category, and construct an essentially surjective functor. To our knowledge, there is no clear standard definition for morphisms between classical CA, so we formulate the classical CA as a discrete category:

Definition 3.15: Classical CA as a category

The category $\mathsf{classical}$ has as objects classical CA according to Definition 2.13, i.e., the collection of quadruples $(\mathscr{M},N,S,\mathring{\gamma})$ where $\mathscr{M}$ is a monoid, $N \subseteq \mathscr{M}$ with $\mathbf{1} \in N$, $S \in \mathbf{Set}$ and $\mathring{\gamma}\colon S^N \to S$. As morphisms it has only identity morphisms.

We will need the following auxiliary lemma in order to establish essential surjectivity:

Lemma 3.16

For every signature $(\mathscr{M}, N, S)$ there exists a functor

$$G^{(\mathscr{M},N,S)} \colon \mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)} \to \mathbf{Mon}$$

such that for every $\mathcal{X} \in \mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$ there exist a monoid isomorphism

$$\psi_{\mathcal{X}} \colon G^{(\mathscr{M},N,S)}(\mathcal{X}) \xrightarrow{\sim} \mathscr{M}.$$

Proof. We abbreviate $G^{(\mathscr{M},N,S)}$ as G. We first define the action of G on objects, so let $\mathcal{X} = (X, \langle \chi, \gamma \rangle, \mathring{\gamma}, r) \in \mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$ be a coalgebra rooted in r. Note that, by definition of a root (Definition 3.8) and the fact that $(-) \circledast r$ is an injection (Definition 3.10), every $x \in X$ can be written in the form $m_x \circledast r$ for a unique $m_x \in \mathscr{M}$. Then we define $G(\mathcal{X})$ to be the monoid $(X, \bullet_X, r)$. Here $\bullet_X$ is defined for every $x = m_x \circledast r \in X$ and $x' = m_{x'} \circledast r \in X$ as

$$(m_x \circledast r) \bullet_X (m_{x'} \circledast r) \stackrel{\text{def}}{=} (m_x \bullet m_{x'}) \circledast r.$$

The monoid identity laws indeed hold for any $m_x \circledast r \in X$:

$$r \bullet_X (m_x \circledast r) = (\mathbf{1} \bullet m_x) \circledast r = m_x \circledast r = (m_x \bullet \mathbf{1}) \circledast r = (m_x \circledast r) \bullet_X \circledast r,$$

where we used the fact that $r = \mathbf{1} \circledast r$. The monoid associativity law follows from the associativity of $\bullet$: let $m_x \circledast r, m_y \circledast r, m_z \circledast r \in X$, then

$$\begin{aligned}
(m_x \circledast r) \bullet_x \big((m_y \circledast r) \bullet_x (m_z \circledast r)\big) &= (m_x \bullet (m_y \bullet m_z)) \circledast r && \text{// Definition } \bullet_X \text{ (used twice).} \\
&= ((m_x \bullet) m_y \bullet m_z) \circledast r && \text{// Associativity of } \bullet. \\
&= \big((m_x \circledast r) \bullet_x (m_y \circledast r)\big) \bullet_x (m_z \circledast r)). && \text{// Definition } \bullet_X \text{ (used twice).}
\end{aligned}$$

Thus $G(\mathcal{X})$ is a well-defined monoid. To define G on morphisms, let $\mathcal{X} = (X, \langle \chi, \gamma \rangle, r_X, \mathring{\gamma})$ and $\mathcal{Y} = (Y, \langle \xi, \delta \rangle, r_Y, \mathring{\delta})$ be coalgebras in $\mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$ and let $h \colon \mathcal{X} \to \mathcal{Y}$ by a coalgebra homomorphism. We claim that we can define $Gh \stackrel{\text{def}}{=} h$, so we show that $h \colon G(\mathcal{X}) \to G(\mathcal{Y})$ is a monoid homomorphism. The first condition of a monoid homomorphism (Definition 2.11) requires preservation of identity elements, but indeed $h(r_X) = r_Y$ by definition of $\mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$. It remains to prove the other condition of a monoid homomorphism. To this end, we first observe that for any $x \in X$, the coalgebra homomorphism law (diagram (6) in Definition 2.18) on h evaluates to:

$$\xi(h(x)) \times (\delta_{h(x)}) = (h \circ \chi(x)) \times (h \circ \gamma_x)$$

Taking the first projections of both sides of the equation gives

$$\xi(h(x)) = h \circ \chi(x),$$

which on input $m \in \mathscr{M}$ evaluates (in infix notation) to:

$$m \mathbin{\text{(}\!\!\text{(}} h(x) = h(m \circledast x). \tag{22}$$

This property allows to prove the second condition of a monoid homomorphism, that $h(x \bullet_X x') = h(x) \bullet_Y h(x')$ for all $x, x' \in X$. Write $x = m_x \circledast r_X$ and $x' = m_{x'} \circledast r_X \in X$, then it indeed holds that:

$$\begin{aligned}
h((m_x \circledast r_X) \bullet_X (m_{x'} \circledast r_X)) &= h((m_x \bullet m_{x'}) \circledast r_X) && \text{// Definition } \bullet_X. \\
&= (m_x \bullet m_{x'}) \mathbin{\text{(}\!\!\text{(}} h(r_X) && \text{// Eq. (22).} \\
&= (m_x \mathbin{\text{(}\!\!\text{(}} h(r_X)) \bullet_Y (m_{x'} \mathbin{\text{(}\!\!\text{(}} h(r_Y)) && \text{// Definition } \bullet_Y. \\
&= h(m_x \circledast r_X) \bullet_Y h(m_{x'} \circledast r_X). && \text{// Eq. (22).}
\end{aligned}$$

That G preserves composition and identity morphisms is trivial, thus we established that G is a functor.

It remains to give the monoid isomorphism $\psi_{\mathcal{X}} \colon G(\mathcal{X}) \xrightarrow{\sim} \mathscr{M}$. In the following, we abbreviate $\psi_{\mathcal{X}}$ by ψ. For all $m \circledast r \in X$, we define:

$$\psi_{\mathcal{X}}(m \circledast r) \stackrel{\text{def}}{=} m$$

This is well-defined since $(-) \circledast r$ is an injection (by Definition 3.10). The monoid homomorphism conditions (Definition 2.11) are straightforward to check:

$$\begin{aligned} \psi(r) &= \psi(\mathbf{1} \circledast r) = \mathbf{1} \\ \psi((m_x \circledast r) \bullet_X (m_{x'} \circledast r)) &= \psi((m_x \bullet m_{x'}) \circledast r) \\ &= m_x \bullet m_{x'} \\ &= \psi(m_x \circledast r) \bullet \psi(m_{x'} \circledast r), \end{aligned}$$

for all $m_x \circledast r, m_{x'} \circledast r \in X$. The inverse of ψ is $(-) \circledast r$, which also satisfies the monoid homomorphism conditions:

$$\begin{aligned} \mathbf{1} \circledast r &= r \\ (m \bullet n) \circledast r &= (m \circledast r) \bullet_X (n \circledast r). \end{aligned}$$

That $((-) \circledast r) \circ \psi = \mathsf{Id}_{G(\mathcal{X})}$ and $\psi \circ ((-) \circledast r) = \mathsf{Id}_{\mathscr{M}}$ are, respectively, easy to see:

$$\begin{aligned} \psi(m \circledast r) \circledast r &= m \circledast r \\ \psi(m \circledast r) &= m. \end{aligned}$$

□

Theorem 3.17

There exists an essentially surjective functor $F \colon \mathsf{classical} \to \mathbf{CCA}_{\mathsf{U}}^{+}$.

Proof. We first construct a functor $F \colon \mathsf{classical} \to \mathbf{CCA}_{\mathsf{U}}^{+}$, and thereafter show that it is essentially surjective.

Given a classical CA $\mathfrak{A} = (\mathscr{M}, N, S, \mathring{\gamma}) \in \mathsf{classical}$, we define $F(\mathfrak{A})$ to be the coalgebra $(\mathscr{M}, \langle \chi, \gamma \rangle, \mathbf{1}, \mathring{\gamma})$, which uses the monoid elements as cells and $\mathbf{1}$ as root. It remains to define χ and γ, and that $F(\mathfrak{A}) \in \mathbf{CCA}_{\mathsf{U}}^{+}$. For the latter, it suffices to show that $F(\mathfrak{A}) \in \mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$, thus we will show that $F(\mathfrak{A})$ satisfies the conditions of $\mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$ as given in Definition 3.10.

For all $m, x \in \mathscr{M}$, we define:

$$m \circledast x \stackrel{\text{def}}{=} m \bullet x.$$

We can immediately confirm that condition 4 of Definition 3.10 is satisfied: the properties of a cowriter comonad coalgebra, Eq. (11) and (12), follow directly from the monoid identity and associativity laws on $\bullet$ (see Definition 2.3).

For all $x \in \mathscr{M}$ and $k \colon \mathscr{M} \to S$, we define

$$\gamma_x(k) \stackrel{\text{def}}{=} \mathring{\gamma}(k \restriction N).$$

Condition 1 of Definition 3.10 holds by construction. Condition 2, which requires that that $\mathbf{1}$ is indeed the root of the CA, follows from the monoid identity law (see Definition 2.3): $m = m \bullet \mathbf{1}$ for all $m \in \mathscr{M}$. Finally, Condition 3, i.e., injectivity of $(-) \circledast \mathbf{1}$, also follows from the monoid idenity law since $\circledast = \bullet$ (for any $m, n \in \mathscr{M}$, if $m \bullet \mathbf{1} = n \bullet \mathbf{1}$ then the monoid identity law gives $m = n$).

This established F as a well-typed functor. As for essential surjectivity, assume any arbitrary $\mathcal{X} = (X, \langle \chi, \gamma \rangle, \mathring{\gamma}, r) \in \mathbf{CCA}_{\mathsf{U}}^{+}$. It is to show that there exists an $\mathfrak{A} \in \mathsf{classical}$ such that $F(\mathfrak{A}) \cong \mathcal{X}$. Without loss of generality, we may assume that $\mathcal{X} \in \mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$ for a signature $(\mathscr{M}, N, S)$.

From the definition of F we can already see that $\mathfrak{A}$ needs to be defined on a monoid with the set X as elements. Using the functor $G^{(\mathscr{M},N,S)}$ of Lemma 3.16, we can choose

$$\mathfrak{A} \stackrel{\text{def}}{=} (G^{(\mathscr{M},N,S)}(\mathcal{X}), N \otimes r, S, \mathring{\gamma}') \in \mathsf{classical},$$

where

$$\mathring{\gamma}' \stackrel{\text{def}}{=} \lambda k \,.\, \mathring{\gamma}(k \circ \psi_{\mathcal{X}})$$

(here $\psi_{\mathcal{X}} \colon G^{(\mathscr{M},N,S)}(\mathcal{X}) \xrightarrow{\sim} \mathscr{M}$ is the isomorphism from the proof of Lemma 3.16 and the type of k is $(N \otimes r) \to S$). The F-image of $\mathfrak{A}$ is

$$F(\mathfrak{A}) = (X, \langle \bullet_X, \gamma' \rangle, \mathring{\gamma}', r) \in \mathbf{CCA}_{\mathsf{U}}^{(G(\mathcal{X}), N \otimes r, S)}$$

where

$$\gamma'_x = \lambda k \,.\, \mathring{\gamma}'(k) = \lambda k \,.\, \mathring{\gamma}(k \circ \psi_{\mathcal{X}}). \tag{23}$$

It remains to show that $F(\mathfrak{A})$ is isomorphic to $\mathcal{X}$. The coalgebras $F(\mathfrak{A})$ and $\mathcal{X}$ are defined on different signatures but with the same state set, so the only possible isomorphism in one of the extra arrows in $\mathbf{CCA}_{\mathsf{U}}^{+}$ (Definition 3.13) that is defined by a monoid isomorphism. And indeed, the monoid isomorphism $\psi_{\mathcal{X}}$ of the proof of Lemma 3.16 satisfies the conditions listed in Definition 3.13:

1. The first condition is immediate.
2. The second condition requires showing that $x \bullet_X r = \psi_{\mathcal{X}}(x) \otimes r$ for all $x \in X$. Recall from the proof of Lemma 3.16 that we can write $x = m \otimes r$ for a unique $m \in \mathscr{M}$, and observe that $r = \mathbf{1} \otimes r$. Hence we can indeed compute that:

$$\begin{aligned} x \bullet_X r &= (m \otimes x) \bullet_X (\mathbf{1} \otimes r) \\ &= (m \bullet \mathbf{1}) \otimes r && \text{// Definition } \bullet_X \text{ (see the proof of Lemma 3.16).} \\ &= m \otimes r \\ &= \psi_{\mathcal{X}}(m \otimes r) \otimes r && \text{// Definition } \psi_{\mathcal{X}} \text{ (see the proof of Lemma 3.16).} \\ &= \psi_{\mathcal{X}}(x) \otimes r. \end{aligned}$$

3. The last condition is exactly Eq. (23).

Hence we indeed have an isomorphism $(\psi_{\mathcal{X}})_{F(\mathfrak{A}),\mathcal{X}} \colon F(\mathfrak{A}) \xrightarrow{\sim} \mathcal{X}$ by Definition 3.13. □

We can also define global rules for coalgebras, in such a way that the global rules for coalgebras in $\mathbf{CCA}_{\mathsf{U}}$ also correspond to the classical global rules.

Definition 3.18: Global rule (coalgebraic)

Let $(\mathscr{M}, N, S)$ be a signature and let Let $\mathcal{X} = (X, \langle \chi, \gamma \rangle) \in \mathsf{Coalg}(C_{\mathsf{U}}^{(\mathscr{M},S)})$. A *configuration* of $\mathcal{X}$ is a function $c \colon X \to S$. The *global rule* of $\mathcal{X}$ is a map between configurations defined as

$$\begin{aligned} G_{\mathcal{X}} &\colon S^X \to S^X \\ G_{\mathcal{X}}(c)(x) &\stackrel{\text{def}}{=} \gamma_x(\lambda m \in \mathscr{M} \,.\, c(m \otimes x)). \end{aligned} \tag{24}$$

Note that for coalgebras $\mathcal{X} = (X, \langle \chi, \gamma \rangle, \mathring{\gamma}, r) \in \mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$ this definition evaluates to:

$$G_{\mathcal{X}}(c)(x) = \mathring{\gamma}(\lambda n \in N \,.\, c(n \circledS x)).$$

Every initial configuration $c_0 \colon X \to S$ induces a *trace sequence* $(c_i)_{i\in\mathbb{N}}$ defined as $c_i \stackrel{\text{def}}{=} G^i_{\mathcal{X}}(c_0)$.

The functor F of Theorem 3.17 preserves global rules exactly:

Lemma 3.19

Let $\mathfrak{A} = (\mathscr{M}, N, S, \mathring{\gamma})$ be a classical CA (as in Definition 2.13), and let $F(\mathfrak{A}) = (\mathscr{M}, \langle \chi, \gamma \rangle, \mathring{\gamma}, \mathbf{1})$ be the output of the functor constructed in Theorem 3.17. Then: $G_{\mathfrak{A}} = G_{F(\mathfrak{A})}$.

Proof. Take any arbitrary configuration $c \colon \mathscr{M} \to S$ and any cell $m \in \mathscr{M}$. Then the classical definition of a global rule (Definition 2.16) shows that

$$G_{\mathfrak{A}}(c)(m) \stackrel{\text{def}}{=} \mathring{\gamma}(\lambda n \,.\, c(n \bullet m)).$$

The coalgebraic global rule (Definition 3.18) of $F(\mathfrak{A})$ is

$$G_{F(\mathfrak{A})}(c)(m) \stackrel{\text{def}}{=} \mathring{\gamma}(\lambda n \,.\, c(n \circledS m)),$$

but by definition of F (see the proof of Theorem 3.17) it holds that $n \circledS m = n \bullet m$, thus the global rules are indeed equal. □

3.4 Modelling irregular and non-uniform CA

We finish this section by providing arguments for further refining the functor C_{U} in the next section in order to better model CA whose cells have distinct local behaviour.

There are two ways in which the local behaviour of cells can be different: (1) they have a different neighbourhood structure, and (2) they can have different local rules. To avoid confusion, I will call the former condition *irregularity* and the latter *non-uniformity*.

Definition 3.20: Irregular CA

A CA is *irregular* if it has two cells x and x' and two paths in the neighbourhood $n_1, n_2 \in N$ such that n_1 and n_2 lead to different cells when traversing them from initial cell x, but to the same cell when traversed from x'.

In case the CA is given as a coalgebra $\mathcal{X} = (X, \langle \chi, \gamma \rangle) \in \mathbf{CCA}_{\mathsf{U}}^{(\mathscr{M},N,S)}$, then this means that $n_1 \circledS x \neq n_2 \circledS x$ but $n_1 \circledS x' = n_2 \circledS x'$.

Example 3.21 (An irregular CA)**.** Let $\mathcal{X} = (\mathbb{Z}, \langle \chi, \gamma \rangle) \in \mathbf{CCA}_{\mathsf{U}}^{(\mathbb{Z}^2,N,S)}$ be the CA with:

- as underlying monoid $(\mathbb{Z}^2, +, (0,0))$,
- neighbourhood $N = \{(i,j) \mid -1 \leq i,j \leq 1\}$,
- and with the following cowriter comonad coalgebra χ:

$$(a,b) \circledS x \stackrel{\text{def}}{=} a + x.$$

Then $\mathcal{X}$ is irregular, since $(0,1),(0,-1) \in N$ but for any $x \in \mathbb{Z}$ it holds that $(0,1) \circledast x = x = (0,-1) \circledast x$.

Irregular CA always have *incomplete* cells:

Definition 3.22: Incomplete cell

An *incomplete* cell is a cell that does not have a distinct neighbour for every $n \in N$.

Remark 3.23. The existence of incomplete cells does not imply irregularity: it is possible that $n_1, n_2 \in N$ lead to the same neighbour for *all* cells in a CA.

Definition 3.24: Non-uniform CA

A CA with neighbourhood N is *non-uniform* if there exist two cells x and x' such that the local rule $\gamma_x \colon S^N \to S$ used to compute the next state of x in a trace sequence is different from the local rule $\gamma_{x'} \colon S^N \to S$ used at x'.

Remark 3.25. We defined *non-uniformity* in such a way that the neighbourhood N is still the same for all cells. This is not restrictive, since one can model a CA where different cells have different neighbourhoods as an *irregular CA*, and take N to be the union of all neighbourhoods.

Both classical definition of a CA (Definition 2.13) and $\mathbf{CCA}^+_{\mathsf{U}}$ allow irregular CA but not non-uniform CA. In particular, all cells in a coalgebra in $\mathbf{CCA}^+_{\mathsf{U}}$ use the same local rule $\mathring{\gamma} \colon S^N \to S$. However, this does *not* imply that all cells behave identically. Furthermore, a uniform CA may have an incomplete cell x, not all inputs of the type S^N to the local rule at x can occur in the computation of a trace sequence. This is because configurations assign states to cells and not to paths. The next example gives a concrete instance of this phenomenon.

Example 3.26. Consider the monoid $\mathscr{M} = (\{\mathbf{1}, A, B\}, \bullet, \varepsilon)$ where

$$A \bullet A = A \qquad B \bullet B = B \qquad A \bullet B = B \bullet A = \mathbf{1}.$$

Define the neighbourhood $N = \{\mathbf{1}, A, B\}$ and the CA $\mathcal{X} = (X, \langle \chi, \gamma \rangle, \mathring{\gamma}, r) \in \mathbf{CCA}^{(\mathscr{M},N,S)}_{\mathsf{U}}$ where $X = \{r, x, y\}$ as in Figure 3.

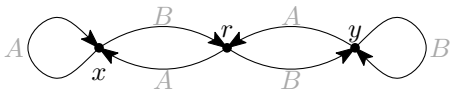


Figure 3: An irregular CA $\mathcal{X} = (\{r, x, y\}, \langle \chi, \gamma \rangle)$ defined on the monoid $\mathscr{M}$ of Example 3.26. Arrows show the action of χ; arrows for the monoid identity element have been omitted. Under the assumption that γ is well-behaved, we see that $\mathcal{X}$ is in $\mathbf{CCA}^+_{\mathsf{U}}$, since it satisfies the conditions of Definition 3.10: it is rooted in r, and $(-) \circledast r$ is an injection since r has 3 distinct neighbours (including itself) while $\mathscr{M}$ has only 3 elements.

Observe that x has only a B-neighbour that is not itself, and y only an A-neighbour. In any configuration $c_0 \colon X \to S$, it will always hold that $c_0(x) = c_0(A \circledast x)$ and $c_0(y) = c_0(B \circledast y)$. When using the global rule $G_{\mathcal{X}} \colon S^X \to S^X$ to compute the next state for x in a trace, the inputs $k \colon N \to S$ to the local rule $\mathring{\gamma}$ are always of the form $\lambda n \,.\, c_0(n \circledast x)$. These will therefore never have $k(\mathbf{1}) \neq k(A)$, but $k(\mathbf{1}) \neq k(B)$ is possible. Similarly for y, the inputs will never have $k(\mathbf{1}) \neq k(B)$, while $k(\mathbf{1}) \neq k(A)$ is possible. Thus only a strict subset of the domain S^N of $\mathring{\gamma}$ will be used at x, which means that the local rule at x contains superfluous data. Even stronger, the reachable part of the domain S^N of $\mathring{\gamma}$ at y is different from that at x, and the intersection are only the constant inputs. That is, x and y behave almost independently, except that they behave the same in the case where there only neighbour has the same state as themselves.

The next section will show how to construct CA such that the local rule at every cell does not contain superfluous data. We will adapt the type of the local rule at a cell x to S^{N_x}, where N_x has only one element for every *distinct* neighbour of x. A side effect of allowing different cells to have different local rules is that it becomes possible to formulate constraints on how the local rules should relate to each other, and the next section will discuss two choices of such constraints in depth.

We only required CA in $\mathbf{CCA}^+_{\mathsf{U}}$ to be rooted in order to prove Theorem 3.17: otherwise the functor $\mathsf{classical} \to \mathbf{CCA}^+_{\mathsf{U}}$ would not be essentially surjective. The latter worry is irrelevant when allowing non-uniform CA, since this implies intentionally including coalgebras not corresponding to classical CA as described in Definition 2.13. Note that the root is also unrelated to the local behaviour of a CA.

3.5 Summary of uniform CA

We have described a coalgebraic model of CA, that exploits the idea that cells and connections between cells can be separate mathematical objects. Concretely, we first modelled a CA's graph structure as comonad coalgebras of the cowriter comonad P (Definition 3.1), parametrised by a monoid of "paths" between cells. We interpreted the carrier sets of those coalgebras as the set of cells. To endow coalgebras with local rules, we further modified the cowriter comonad P to a "uniform CA" functor C_{U}. From the coalgebras of C_{U} we constructed the category $\mathbf{CCA}^+_{\mathsf{U}}$, whose objects correspond to the classical, uniform CA (Definition 2.13) as coalgebras: Theorem 3.17 shows that there exists an essentially surjective functor from the classical (uniform) CA to $\mathbf{CCA}^+_{\mathsf{U}}$. Finally, we have discussed irregular and non-uniform CA, and sketched why we will refine C_{U} in the next section in order to model such CA more naturally.

4 General cellular automata

In this section we will refine C_{U} to a functor C_{G}, whose coalgebras can also be non-uniform CA. We will define the class of *general CA*, contains not only the class of classical CA but also non-uniform CA and CA without a root. Thereafter we show how the class of general CA and two subclasses thereof can be specified as subcategories of $\mathsf{Coalg}(C_{\mathsf{G}})$.

Aside from capturing a more general definition of CA, the C_{G}-coalgebras will also have more appropriately typed local rules for irregular CA. Recall from Section 3.4 (and in particular, Example 3.26) that coalgebras of irregular CA in $\mathbf{CCA}^+_{\mathsf{U}}$ have redundant data in their local rules, as these local rules take assignments $N \to S$ of states to paths in the neighbourhood N. In the computation of a trace sequence, states are assigned to cells instead, so not all assignments $N \to S$ can occur in this computation at an incomplete cell x. The functor C_{G} instead will ensure, at type level, that local rule of a coalgebra at a cell x takes as input an assignment of one state to each *distinct* neighbour of x.

4.1 Classes of general CA

When allowing cells in a CA to have different local rules, the question arises whether there should be a correspondence between the distinct local rules of distinct cells, if any at all. There is no unique answer to this question, and different applications of CA may require different degrees of uniformity. In particular, we consider the following hierarchy classes of CA with various degrees of uniformity, from most general to most specific:

1. **All general CA:** the largest class allows all cells to have unrelated local rules.
2. **Neighbourhood uniform (general) CA:** cells with the same neighbourhood structure have the same local rule. This will be more clearly defined in Definition 4.8.

3. **Extension uniform (general) CA:** the local rule of a cell is an extension of the local rule of cells whose neighbourhood is smaller. Cells with the same neighbourhood structure have the same local rule. This will be more clearly defined in Definition 4.8.

4. **Uniform general CA:** all cells have essentially the same local rule $\mathring{\gamma}\colon S^N \to S$, but only with a local modification to the local neighbourhood of the cell. This is simple to implement: given an assignment $k\colon (N \circledast x) \to S$ of states to the neighbours of x, one can simply use $\gamma_x(k) = \mathring{\gamma}(\lambda n \in N \,.\, n \circledast x)$. Hence this class is similar to $\mathbf{CCA}^+_{\mathsf{U}}$ (Definition 3.10), except for (1) the domain of the local rule at a cell x is adapted the neighbourhood structure of x (2) not requiring the existence of a root r nor injectivity of $(-) \circledast r$.

One may need an infinite amount of data to describe an arbitrary general CA, even when using a finite state set and a finite neighbourhood, since the local rule of every individual cell must be specified separately. However, both in the neighbourhood-uniform and extension-uniform classes, the number of distinct local rules on a neighbourhood N is bounded by $2^{\mathsf{card}(N)}$. Furthermore, the appropriate local rule of a cell does not need to be given individually for every cell: it can be determined locally from observing the neighbourhood structure of the cell.

More classes can be defined, but are not considered in this thesis. For example, there is a class of CA whose local rules to agree on inputs that assign every neighbour the same state.

The next step is to formally define the class of general CA, as well as the subclasses of neighbourhood-uniform CA and extension-uniform CA. Before we can give the definitions, we first need to clarify the notion of neighbourhood structures and how they can be compared. After defining these three classes of CA, we will continue this section with showing how each can be modelled as a subcategory of the coalgebras of a "general CA" functor C_{G}, which is a modified version of C_{U}. To this end, we will describe how the relations that neighbourhood-uniformity and extension-uniformity enforce between local rules of a coalgebra can be formulated as naturality conditions on the local rule component γ of the coalgebra's dynamics.

Neighbourhood quotients The definition of a general CA takes a more abstract view on the neighbourhood of a cell x, not depending on the neighbours $x' \in N \circledast x$ as cells, but on rather on the set N_x of sets of paths from x to distinct neighbours. This perspective makes it possible to compare different cells that have essentially the same neighbourhood lay-out, even when the actual the neighbours (as elements of X) are distinct.

Definition 4.1: Neighbourhood quotients

For a cell $x \in X$, define the *neighbourhood quotient* $N_x \stackrel{\text{def}}{=} N/\sim_x$ where $\sim_x \subseteq N \times N$ is the relation defined for all $n, m \in N$ as:

$$(n \sim_x m) \Leftrightarrow (n \circledast x = m \circledast x). \tag{25}$$

We denote the equivalence class (with respect to x) of a path $n \in N$ by $|n|_x \in N_x$.

We typically use the abbreviation *neighbourhood* for *neighbourhood quotient*.

Observe that N_x is always a partitioning of N (Definition 2.1). Consequently, the poset structure on partitionings (Definition 2.2) provides a means of comparing neighbourhoods:

Definition 4.2: Compatible neighbourhood-configuration

Let $N_x, N_{x'} \in \mathsf{PartOrd}(N)$ be the neighbourhoods of cells $x, y \in X$. Then N_x is N_y-*compatible* if $N_y \leq N_x$.

A neighbourhood configuration $z\colon N \to S$ is N_x-*compatible* if

$$\forall_{n,m\in N}\left[|n|_x = |m|_x \Rightarrow z(n) = z(m)\right]. \tag{26}$$

Intuitively: z is N_x-compatible if for all paths n and m (starting from x) that lead to the same cell y, then z assigns these paths the same state.

Remark 4.3. The condition $N_{x'} \leq N_x$ implies

$$\bigwedge_{n,m\in N}\left[|n|_x = |m|_x \Rightarrow |n|_{x'} = |m|_{x'}\right].$$

It follows that any $N_{x'}$-compatible $z\colon N \to S$ is also N_x compatible when $N_{x'} \leq N_x$ (since we would have $|n|_x = |m|_x \Rightarrow |n|_{x'} = |m|_{x'} \Rightarrow z(n) = z(m)$).

The main purpose of the concept of compatibility is allowing us to "downscale" neighbourhood configurations $N \to S$ into the shape of the neighbourhood N_x of a particular cell x:

Definition 4.4: Downscaling

Given an N_x compatible neighbourhood configuration $z\colon N \to S$,
define the x-*downscaled* function $z\!\!\downarrow\!\!x\colon N_x \to S$ as:

$$(z\!\!\downarrow\!\!x)(|n|_x) \stackrel{\text{def}}{=} z(n).$$

Note that this definition is well-defined (independent of the choice of the representative of $|n|_x$): this follows directly from the definition of N_x-compatibility (Definition 4.2).

Example 4.5 (Irregular neighbourhood structures)**.** The CA of Figure4 has four cells each with a different neighbourhood:

- For x, the three neighbourhood paths in $N = \{\varepsilon, L, D\}$ all lead to a different cell. So $N_x = \{|\varepsilon|, |L|, |D|\}$.
- For y, both ε and L lead back to itself, but D leads to z. Thus $N_y = \{|\varepsilon, L|, |D|\}$.
- For w, both ε and D lead back to itself, but D to z. Therefore $N_w = \{|\varepsilon, D|, |L|\}$.
- For z, all paths in N lead back to itself. So $N_z = \{N\}$.

(here we denoted entire equivalence classes as a set delimited with $|(-)|$, a convention we will continue to use when explicitly giving neighbourhood-quotient equivalence classes). In particular, note that y and w have the same number of neighbours, but still a different neighbourhood structure. In fact, N_y and N_w are incomparable in $\mathsf{PartOrd}(N)$. However, we do have the relations $N_z \leq N_x$, $N_y \leq N_x$, $N_w \leq N_x$, $N_z \leq N_y$ and $N_z \leq N_w$. Note that all neighbourhoods are N_z-compatible.

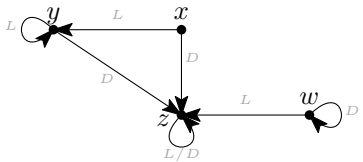


Figure 4: Example of a non-uniform CA with four cells $X = \{w, x, y, z\}$ on the monoid freely generated over the alphabet $\{L, D\}$ and neighbourhood $N = \{\varepsilon, L, D\}$, where ε (the empty string) is the monoid identity. Note that all cells have a different neighbourhood structure.

Suppose we have state set $S = \{\bullet, \circ\}$ and a neighbourhood configuration $k\colon N \to S$ such that $k(\varepsilon) = k(L) = \circ$ and $k(D) = \bullet$. Then k is compatible with N_x and N_y, but not with N_z or N_w.

Classes of generalised CA The definition of a general CA is very similar to that of a classical CA (Definition 2.13), except that it assigns every cell its own local rule (allowing non-uniformity) and allows CA without a root. The subclasses of neighbourhood-uniform and extension-uniform CA follow the same definition, but with additional constraints on the local rule. The presented form of the definition of a general CA is designed to facilitate modelling it coalgebraically:

Definition 4.6: General CA

Static structure
A general CA is a tuple $\mathbb{X} = (\mathscr{M}, N, S, X, \chi, \gamma)$ where:

- $(\mathscr{M}, N, S)$ is a signature (Definition 3.7).
- $X \in \mathbf{Set}$ is a set of *cells*.
- (X, χ) is a cowriter-comonad coalgebra (see Definition 3.1 and 3.2). Equivalently, $\chi\colon X \to X^{\mathscr{M}}$ is an action of $\mathscr{M}$ that satisfies Eq. (11) and (12).
- γ is a family of maps $\gamma_x\colon S^{N_x} \to S$ indexed by $x \in X$ (called the *local rule*).

Dynamics
A *configuration* of $\mathbb{X}$ is an assignment $c\colon X \to S$. Given an *initial configuration* c_0, the *trace sequence* of $\mathbb{X}$ on c_0 is the sequence of configurations $(c_i)_{i\in\mathbb{N}}$, where for every $i \in \mathbb{N}$, c_i is computed inductively as:

$$c_{t+1} \stackrel{\text{def}}{=} G_{\mathbb{X}}^{t+1}(c_0). \tag{27}$$

Here the *global rule* $G_{\mathbb{X}}\colon S^X \to S^X$ is defined as:

$$G_{\mathbb{X}}(c_t)(x) \stackrel{\text{def}}{=} \gamma_x(\lambda\, |n|_x \in N_x \,.\, c_t(n \mathbin{⩒} x)). \tag{28}$$

The class of neighbourhood-uniform CA is the subclass of general CA (Definition 4.6) of CA with the following property: cells with the same neighbourhood use the same local rule. The subclass of extension-uniform CA only contains general CA with the following property: the local rules of cells with large neighbourhoods "extend" the behaviour of cells with smaller neighbourhoods. Extensions are defined as follows:

Definition 4.7: Extension

Let X and Y be sets of cells, and let $\gamma\colon X \to S^{S^N}$ and $\delta\colon Y \to S^{S^N}$ be local rules on the same signature $(\mathscr{M}, N, S)$. A component $\gamma_x\colon S^{N_x} \to S$ at a cell $x \in X$ *extends* the component $\delta_y\colon S^{N_y} \to S$ at a cell y if $N_y \leq N_x$ and for all N_y-compatible $z\colon N \to S$ it holds that

$$\gamma_x(z \mathbin{⩓} x) = \delta_y(z \mathbin{⩓} y).$$

Definition 4.8: Subclasses of general CA

Let $\mathbb{X} = (\mathscr{M}, N, S, X, \chi, \gamma)$ be a general CA.

Then $\mathbb{X}$ is:

- *neighbourhood-uniform* if if it has the property that $\gamma_x = \gamma_{x'}$ when $N_x = N_{x'}$ (for all $x, x' \in X$).
- *extension-uniform* if if for all $x, x' \in X$ with $N_{x'} \leq N_x$ it holds that γ_x extends $\gamma_{x'}$ (according to Definition 4.7).

Remark 4.9. The class of extension-universal is a subclass of the class of neighbourhood-universal CA, which can be seen as follows. Suppose that $\mathbb{X}$ is extension-uniform, then $\mathbb{X}$ is also neighbourhood-uniform if and only if for all $x, x' \in X$ such that $N_x = N_{x'}$ it holds that $\gamma_x = \gamma_{x'}$. But if $N_x = N_{x'}$, then also $N_{x'} \leq N_x$, thus by (4.7) we find

$$\gamma_x(z \diagdown\!\diagdown x) = \gamma_{x'}(z \diagdown\!\diagdown x') \tag{29}$$

for all $z\colon N \to S$ that are $N_{x'}$-compatible. But z is $N_{x'}$-compatible if and only it is N_x-compatible, so all inputs to γ_x and $\gamma_{x'}$ are of the form $z \diagdown\!\diagdown x$ and $z \diagdown\!\diagdown x'$, respectively, for N_x-compatible $z\colon N \to S$. Therefore (29) fully specifies γ_x and $\gamma_{x'}$, and we indeed conclude that $\gamma_x = \gamma_{x'}$.

The next two examples illustrate the consequences of neighbourhood- and extension-uniformity on two specific CA graphs.

Example 4.10 (Forktree)**.** Define the "forktree" $\mathscr{F}$ as the tree where the branching factor of nodes at an even depth is 2, but 1 for nodes at an odd depth. More precisely, $\mathscr{F}$ is the monoid generated (Definition 2.8) over the alphabet $\{L, R\}$ and the following equalities between elements:

$$sL = sR \iff \exists_{n \in \mathbb{N}} [s \in \{L, R\}^{2n+1}], \tag{30}$$

that is, the L- and R-successors of odd-length strings are the same element.

See Figure 5 for a visualisation of $\mathscr{F}$.

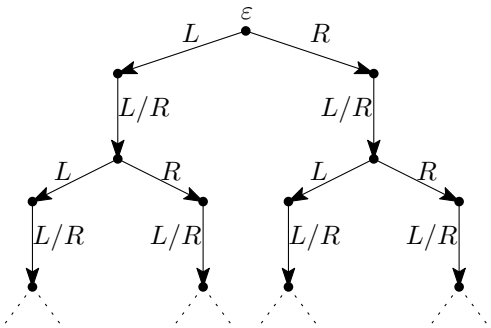


Figure 5: The forktree monoid. The top point is the empty string ε. Labels of arrows denote the neighbourhood element that need to be right-multiplied with the tail point to obtain the head point; the notation "L/R" is used in case multiplication with L and R lead to the same result. Arrows with label ε (which are all self-loops) have been omitted. Observe that some points have the same L- and R-successor, while other points have two distinct successors. Only the generating elements $\{L, R\}$ are shown as arrows, composite arrows are omitted.

In this example we consider general CA of the form $\mathbb{X} = (\mathscr{F}, N, S, X, \langle \chi, \gamma \rangle)$ whose graph is exactly the forktree, and whose neighbourhood is the direct-successor relation:

- $N = \{\varepsilon, L, R\}$.
- $X = \mathscr{F}$.

- $\circledast$ is the forktree's multiplication.

Observe that this CA is irregular: there are two classes of cells based on their neighbourhood quotient. The first class $X_1 \subseteq X$ are the cells at an odd depth in the tree (containing cells such as R, L, RRR, LLL, etc.). By the equalities of the forktree Eq. (30), these cells have branching factor 1 and therefore have the same L- and R-successor. Their neighbourhood quotient is:

$$N_R = \{|\varepsilon|, |L, R|\}.$$

The other class, X_2, consists of cells at even depth with branching factor 2: they have distinct L- and R-successors. These cells have neighbourhood quotient

$$N_\varepsilon = \{|\varepsilon|, |L|, |R|\} \cong N.$$

This class contains cells such as ε, LL, RR, $LLRR$, etc.

Neighbourhood-uniformity

If we want $\mathbb{X}$ to be *neighbourhood-uniform*, then all cells in X_1 need to have the same local rule, and all cells in X_2 need to have the same local rule as well. Thus we would be free to independently choose two local rules $\gamma_1, \gamma_2 \colon X \to S^{S^N}$ and set, for every $x \in X$:

$$\gamma_x = \begin{cases} \gamma_1 & x \in X_1 \\ \gamma_2 & x \in X_2 \end{cases}$$

For example, we may take $S = \{\bullet, \circ\}$ and define γ the following rule set:

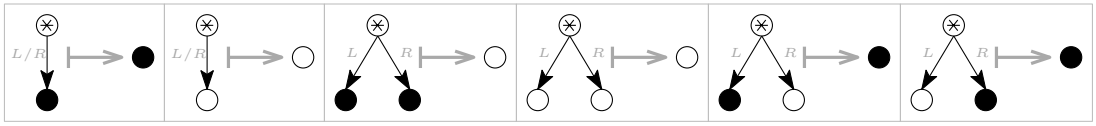


Figure 6: Example definitions for γ_1 (left two rules) and γ_2 (right four rules). The $\star$ denotes that a copy of the rule holds for every possible value at that cell.

An example of a possible trace sequence with this local rule is the following:

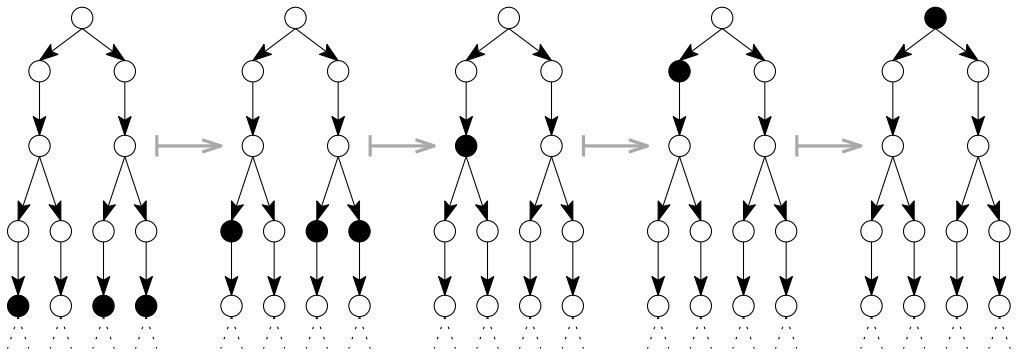


Figure 7: Example trace sequence on the forktree graph $\mathscr{F}$ using the local rules of Figure 6 and an arbitrary initial configuration $\mathscr{F} \to \{\bullet, \circ\}$ (the leftmost configuration). Cells that are not drawn are all in state $\circ$.

Extension-uniformity

If we instead want $\mathbb{X}$ to be *extension-uniform*, then we need to take the fact that $N_R \leq N_\varepsilon$ into account.

This means that we can choose γ_1 freely, but γ_2 must extend γ_1: for every neighbourhood configurations $z\colon N \to S$ such that $z(L) = z(R)$, we need $\gamma_2(z\!\!\setminus\!\!\varepsilon) = \gamma_1(z\!\!\setminus\!\!R)$. The outputs of γ_2 on inputs with $z(L) = z(R)$ can still be chosen independently of γ_1.

The local rule γ presented above in Figure 6 is not extension-uniform. In particular, the following combination of local rules violates the extension property, since the right neighbourhood configuration $N_\varepsilon \to S$ is an "upscaled" version of the left configuration $N_R \to S$, and should therefore give the same output (which is $\circ$):

Example 4.11 (Three-forktree)**.** Define the three-forktree monoid $3\mathscr{F}$ as the monoid generated over the alphabet $\{L, M, R\}$ and with equalities:

$$\begin{aligned} s_1 M s_2 R &= s_1 M s_2 M \\ s_1 M s_2 L &= s_1 M s_2 M \\ s_1 R s_2 L &= s_1 R s_2 M \\ s_1 L s_2 R &= s_1 L s_2 M \end{aligned}$$

for all $s_1, s_2 \in \{L, M, R\}^*$. Figure 8 sketches this monoid.

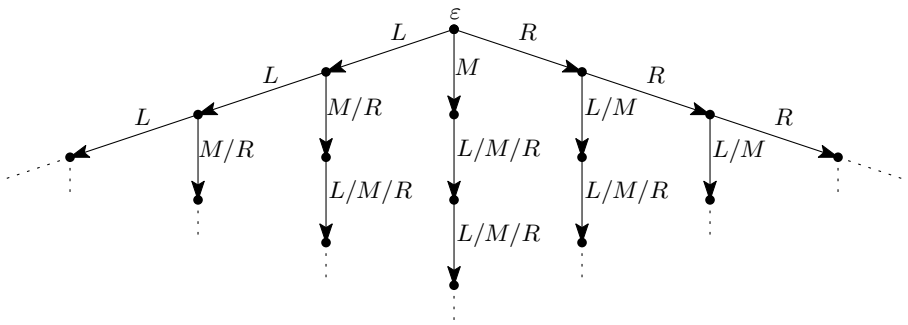


Figure 8: Visualisation of (a part of) $3\mathscr{F}$ with the points $a = \varepsilon$, $x = L$, $y = RR$ and $w = M$ annotated.

As in Example 4.10, we consider CA $\mathbb{X} = (\mathscr{F}, N, S, X, \langle \chi, \gamma \rangle)$ whose graph structure correspond exactly to the monoid structure of $3\mathscr{F}$, and whose neighbourhood is the direct-successor relation:

- The neighbourhood is $N = \{\varepsilon, L, M, R\}$.
- $X = 3\mathscr{F}$.
- χ is $3\mathscr{F}$'s multiplication.

Now consider the following three neighbourhood configurations $z_i\colon N \to S$ with $N \stackrel{\text{def}}{=} \{\varepsilon, L, M, R\}$ and $S \stackrel{\text{def}}{=} \{s_0, s_1, s_2, s_3\}$:

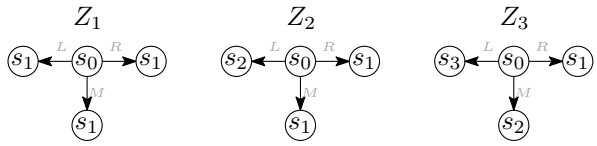


Figure 9: Three neighbourhood configurations $z_1, z_2, z_3 \colon N \to S$ defined on $3\mathscr{F}$ (see Figure 8) with neighbourhood $N \stackrel{\text{def}}{=} \{\varepsilon, L, M, R\}$ and state-set $S \stackrel{\text{def}}{=} \{s_0, s_1, s_2, s_3\}$. All three map ε to s_0, as indicated by the central nodes.

From Figure 8 it becomes clear that there are four classes of cells with the same neighbourhood quotient:

- $X_\varepsilon = \{\varepsilon\}$: ε is the unique cell with neighbourhood $N_\varepsilon \cong N$.
- X_L: the cells $\{L, LL, LLL, \dots\}$ with neighbourhood $N_L = \{|\varepsilon|, |L|, |M, R|\}$.
- X_R: the cells $\{R, RR, RRR, \dots\}$ with neighbourhood $N_R = \{|\varepsilon|, |L, M|, |R|\}$.
- X_M: all other cells (so this includes M, MM, RM, LMM, etc.). These all have neighbourhood $N_M = \{|\varepsilon|, |L, M, R|\}$.

Note that $N_L \neq N_R$, despite being isomorphic as sets. Even stronger, $N_L \not\leq N_R$ and $N_R \not\leq N_L$: the neighbourhood quotients are incomparable in $\mathsf{PartOrd}(N)$.

Neighbourhood-uniformity
If we want $\mathbb{X}$ to be *neighbourhood-uniform*, then we can independently define a local rule γ_i for every choice of $i \in \{\varepsilon, L, R, M\}$, with the requirement that $\gamma_x = \gamma_i$ whenever $x \in X_i$.

Extension-uniformity
If instead we want $\mathbb{X}$ to be *extension-uniform*, then we need to take into account that

$$N_M \leq N_L \leq N_\varepsilon$$

and

$$N_M \leq N_R \leq N_\varepsilon.$$

Therefore γ_L and γ_R only need to extend γ_M and not need to extend each other (since N_L and N_R are incomparable). However, γ_ε needs to extend γ_L and γ_R (and γ_M, but any extension of γ_L or γ_R always is an extension of γ_M).

One may wonder if the extension requirement of γ_ε restricts the choices of γ_L and γ_R: after all, γ_ε cannot be an extension of two disagreeing local rules at the same time. The answer is negative: we can choose γ_L independently from γ_R, this will not cause problems when choosing γ_ε. To see this, we observe that any $z \colon N \to S$ that is N_L-compatible but not N_M-compatible assigns L and M different states, and is therefore not N_R-compatible. Similarly, if z is N_R-compatible but not N_M-compatible, then z is not N_L compatible. So the only inputs on which both γ_L and γ_R are defined are N_M-compatible, on which γ_L and γ_R agree.

4.2 General CA functor

We will now define the "general CA" functor C_{G}, which is a modification of C_{U} (Definition 3.5) that ensures that the local rule components of coalgebras adapt to the neighbourhoods of cells. The idea is simple: where the γ component of a C_{U}-coalgebra had type $\gamma_x \colon S^{\mathscr{M}} \to S$ for every cell $x \in X$, the γ component of a C_{G}-coalgebra instead has type $\gamma_x \colon S^{N_x} \to S$. Hence it takes inputs of type $N_x \to S$ that assign exactly one state to every distinct neighbour of x.

Coalgebraic models of general CA arise as a full subcategory $\mathbf{CCA}_{\mathsf{G}}$ of $\mathsf{Coalg}(C_{\mathsf{G}})$. Since neighbourhood-uniform CA and extension-uniform CA are subclasses of general CA, we in turn construct coalgebraic models of these subclasses as subcategories of $\mathbf{CCA}_{\mathsf{G}}$.

The following definition defines C_{G}. It is not directly obvious that it satisfies the functor laws, but this will be proven in Lemma 4.15.

Definition 4.12

For every signature $(\mathscr{M}, N, S)$, the functor $C_{\mathsf{G}}\colon \mathbf{Set} \to \mathbf{Set}$ is defined on objects $X \in \mathbf{Set}$ as:

$$C_{\mathsf{G}}(X) \stackrel{\text{def}}{=} \sum_{f \in X^{\mathscr{M}}} S^{S^{\mathsf{Prelm}_N(f)}}. \tag{31}$$

(where $\sum$ denotes the dependent sum of Definition 2.23).

The action of C_{G} on functions $h\colon X \to Y$ is defined as:

$$\begin{aligned} C_{\mathsf{G}}h\colon \sum_{f \in X^{\mathscr{M}}} S^{S^{\mathsf{Prelm}_N(f)}} &\to \sum_{f' \in Y^{\mathscr{M}}} S^{S^{\mathsf{Prelm}_N(f')}} \\ C_{\mathsf{G}}h\colon (f, g) &\mapsto (h \circ f, \lambda k\,.\, g(\lambda\, |n|_f\ .\, k(|n|_{h \circ f}))). \end{aligned} \tag{32}$$

where $|n|_f$ gives the equivalence class of $n \in N$ under the relation $n' \sim n''$ iff $f(n') = f(n'')$ (and analogous for $|n|_{h \circ f}$).

Remark 4.13. As for C_{U}-coalgebras (Definition 3.5), C_{G}-coalgebras (X, d) have two components $\chi(x) \stackrel{\text{def}}{=} \pi_1(d(x))$ and $\gamma_x \stackrel{\text{def}}{=} \pi_2(d(x))$ for every $x \in X$. We continue the convention of omitting the projections and denote the coalgebra directly as $(X, \langle \chi, \gamma \rangle)$.

Remark 4.14. There are three important insights not directly obvious from the definition of C_{G}:

1. For a coalgebra $(X, \langle \chi, \gamma \rangle) \in \mathsf{Coalg}(C_{\mathsf{G}})$ it holds that $\mathsf{Prelm}_N((-) \circledS x) = N_x$ (see Definition 4.1) for every cell $x \in X$. This means that $\gamma_x\colon S^{N_x} \to S$, as desired.
2. The element in $preim_N((-) \circledS x)$ are equivalence classes of the form $|n|_{(-) \circledS x}$ for some $n \in N$. By the previous remark, this class is the same as $|n|_x$.
3. For $h\colon X \to Y$, the map $C_{\mathsf{G}}h$ is well-defined only if $|n|_{h \circ f}$ is independent of the choice of the representative of $|n|_f$. This is indeed true: if $|n|_f = |n|_f$, then $f(n) = f(n')$, which implies $h(f(n)) = h(f(n'))$. The latter implies $|n|_{h \circ f} = |n'|_{h \circ f}$. Note that the type of the bound variable k is $k\colon \mathsf{Prelm}_N(h \circ f) \to S$.

Lemma 4.15

The action of C_{G} (Definition 4.12) on morphisms is functorial.

Proof. Let $X \in \mathbf{Set}$ and $(f, g) \in C_{\mathsf{G}}X = \sum_{f \in X^{\mathscr{M}}} S^{S^{\mathsf{Prelm}_N(f)}}$, then identity preservation is simple to verify:

$$\begin{aligned} C_{\mathsf{G}}\mathsf{Id}_X(f, d) &= (\mathsf{Id}_X \circ f, \lambda k\,.\, d(\lambda x\,.\, k(\mathsf{Id}_X(x)))) \\ &= (f, \lambda k\,.\, d(\lambda x\,.\, k(x))) \\ &= (f, \lambda k\,.\, d(k)) \\ &= (f, d). \end{aligned}$$

Composition preservation follows from evaluating λ-expressions. Take any pair of functions:

$$X \xrightarrow{h} Y \xrightarrow{\ell} Z$$

and an input $(f, g) \in C_{\mathsf{G}}X = \sum_{f \in X^{\mathscr{M}}} S^{S^{\mathsf{PreIm}_N(f)}}$. Then we compute:

$$\begin{aligned}
(C_{\mathsf{G}}\ell \circ C_{\mathsf{G}}h)(f,g) &= C_{\mathsf{G}}\ell(h \circ f, \lambda c\,.\, g(\lambda\, |n|_f\ .\, c(|n|_{h\circ f}))) && \text{// Evaluate } C_{\mathsf{G}}h.\\
&= (\ell \circ h \circ f, \lambda k\,.\, \Big(g(\lambda\, |n|_f\ .\, c(|n|_{h\circ f}))\Big)\Big(\lambda\, |n'|_{h\circ f}\ .\, k(|n'|_{\ell\circ h\circ f})\Big)) && \text{// Evaluate } C_{\mathsf{G}}\ell.\\
&= (\ell \circ h \circ f, \lambda k\,.\, \Big(g(\lambda\, |n|_f\ .\, \Big(\lambda\, |n'|_{h\circ f}\ .\, k(|n'|_{\ell\circ h\circ f})\Big)\,(|n|_{h\circ f}))\Big)) \\
& && \text{// Substitute } \lambda\, |n'|_{h\circ f}\ .\, k(|n'|_{\ell\circ h\circ f}) \text{ for } c.\\
&= (\ell \circ h \circ f, \lambda k\,.\, \Big(g(\lambda\, |n|_f\ .\, k(|n|_{\ell\circ h\circ f}))\Big)) && \text{// Substitute } |n|_{h\circ f} \text{ for } |n'|_{h\circ f}.\\
&\overset{\text{def}}{=} C_{\mathsf{G}}(\ell \circ h)(f,g) && \text{// Definition } C_{\mathsf{G}}(\ell \circ h).
\end{aligned}$$

as required. Note that the bound variables have types

$$c\colon \underset{N}{\mathsf{PreIm}}(h \circ f) \to S$$

and

$$k\colon \underset{N}{\mathsf{PreIm}}(\ell \circ h \circ f) \to S.$$

□

The type of coalgebras of $C_{\mathsf{G}}^{(\mathscr{M},N,S)}$ together with Remark 4.14.1 make it clear that the subcategory of $C_{\mathsf{G}}^{(\mathscr{M},N,S)}$-coalgebras whose first component is a cowriter-comonad coalgebra are exactly the general CA on signature $(\mathscr{M}, N, S)$ (Definition 4.6). Thus we can define general CA coalgebraically as:

Definition 4.16: Category of general CA coalgebras

For every signature $(\mathscr{M}, N, S)$, let $\mathbf{CCA}_{\mathsf{G}}^{(\mathscr{M},N,S)}$ be the subcategory of $\mathsf{Coalg}(C_d^{(\mathscr{M},N,S)})$ of coalgebras whose first component is a cowriter-comonad coalgebra (Definition 3.1 and 3.2). Let the *category of general CA coalgebras*, denoted $\mathbf{CCA}_{\mathsf{G}}$, be the coproduct of $\mathbf{CCA}_{\mathsf{G}}^{(\mathscr{M},N,S)}$ over all signatures $(\mathscr{M}, N, S)$.

4.3 Neighbourhood- and extension-uniform CA

We claimed that the coalgebras $\mathcal{X} = (X, \langle \chi, \gamma \rangle) \in \mathbf{CCA}_{\mathsf{G}}$ that are neighbourhood-uniform or extension-uniform CA respectively, are exactly the coalgebras whose second projection (the local rule γ) satisfies a respective naturality constraint. We will show this via backward reasoning: first we show that neighbourhood- and extension-uniformity can be stated as an commuting diagram involving γ, and thereafter we construct the categories and functors that show that this diagram is a naturality condition.

In the remainder of this section, we fix a general CA $\mathcal{X} = (X, \langle \chi, \gamma \rangle) \in \mathbf{CCA}_{\mathsf{G}}$ on a signature $(\mathscr{M}, N, S)$.

Extension-uniformity First consider extension-uniformity (Definition 4.8), which requires that for any pair of cells $x, x' \in X$ with $N_{x'} \leq N_x$ it holds that γ_x *extends* $\gamma_{x'}$ (extensions were defined in Definition 4.7, and $\leq$ is from the poset $\mathsf{PartOrd}(N)$ of Definition 2.2). That is, for any $N_{x'}$-compatible $z\colon N \to S$ it holds that

$$\gamma_x(z \lVert x) = \gamma_{x'}(z \lVert x'),$$

where $(z \backslash\!\backslash x) \in S^{N_x}$ and $(z \backslash\!\backslash x') \in S^{N_{x'}}$. We can construct a mapping $S^{N_{x'}} \to S^{N_x}$ that sends $z \backslash\!\backslash x'$ to $z \backslash\!\backslash x$ as follows:

$$Q_{x,x'} \colon S^{N_{x'}} \to S^{N_x}$$
$$Q_{x,x'}(k) = \lambda\, |n|_x \;.\; k(|n|_{x'}).$$

This mapping is well-defined (independent of the choice of representative of $|n|_x$), this follows from the definition of $\leq$ on $N_{x'} \leq N_x$. Now indeed we can check that

$$\begin{aligned} Q_{x,x'}(z \backslash\!\backslash x') &\stackrel{\text{def}}{=} \lambda\, |n|_x \;.\; (z \backslash\!\backslash x')(|n|_{x'}) && \text{// Definition } Q_{x,x'}. \\ &= \lambda\, |n|_x \;.\; z(n) && \text{// Definition } \backslash\!\backslash \text{ (Definition 4.4).} \\ &= \lambda\, |n|_x \;.\; (z \backslash\!\backslash x)(|n|_x) && \text{// Definition } \backslash\!\backslash \text{ (Definition 4.4).} \\ &= z \backslash\!\backslash x, \end{aligned}$$

as we desired.

Thus, γ is extension-uniform if, for all $x, x' \in X$ with $N_{x'} \leq N_x$, the following diagram commutes:

$$\begin{array}{ccc} S^{N_{x'}} & \xrightarrow{\gamma_{x'}} & S \\ {\scriptstyle Q_{x,x'}}\downarrow & & \| \\ S^{N_x} & \xrightarrow{\gamma_x} & S \end{array} \tag{33}$$

Neighbourhood-uniformity The reasoning for neighbourhood-uniformity is the same, except with the condition $N_{x'} \leq N_x$ replaced by $N_{x'} = N_x$. To see this, recall from the discussion in Remark (4.9) that, under the assumption $N_{x'} = N_x$, the condition $\gamma_x = \gamma_{x'}$ is equivalent to the condition that γ_x extending $\gamma_{x'}$ (which is equivalent to $\gamma_{x'}$ extending γ_x). Thus, γ is neighbourhood-uniform if and only if Diagram (33) commutes for any pair of cells $x, x' \in X$ such that $N_x = N_{x'}$.

4.3.1 The local rule as a natural transformation

The previous paragraphs showed that γ is neighbourhood- and extension-uniform, respectively, if Diagram (33) commutes whenever $N_{x'} \leq N_x$ or $N_{x'} = N_x$, respectively. To show that this is a naturality condition on γ, we define appropriate categories and functors such that Diagram (33) is a naturality square.

As domain of the functors we use the following thin category, using the poset structure of $\mathsf{PartOrd}(N)$:

Definition 4.17: Neighbourhood category

The *neighbourhood category of* $\mathcal{X}$, denoted as $\mathbf{N}_{\mathcal{X}}$, is the category with as objects the neighbourhood quotients N_x of every $x \in X$, and a unique morphism $N_{x'} \to N_x$ whenever $N_{x'} \leq N_x$.

The *neighbourhood-iso-category of* $\mathcal{X}$, denoted as $\mathbf{N}_{\mathcal{X}}^{=}$, has the same objects as $\mathbf{N}_{\mathcal{X}}$ but only identity morphism.

For extension-uniformity, the first functor is $Q \colon \mathbf{N}_{\mathcal{X}} \to \mathbf{Set}$ that sends every object N_x to S^{N_x}. On a morphism $f \colon N_{x'} \leq N_x$ we define $Qf \stackrel{\text{def}}{=} Q_{x,x'}$, as required by Diagram (33). The second functor is Δ_S that maps every object to the set S, and every morphism to Id_S.

The condition that γ is a natural transformation $\gamma \colon Q \Rightarrow \Delta_S$ means that, for every morphism $f \colon N_{x'} \leq N_x$, Diagram (33) commutes. As argued above, this is exactly the extension-uniformity requirement.

As for neighbourhood-uniformity, we can use the functor $Q^{=}\colon \mathbf{N}^{=}_{\mathcal{X}} \to \mathbf{Set}$, which is defined the same way as Q. The naturality condition $\gamma\colon Q^{=} \Rightarrow \Delta_S$ means that Diagram 33 need to commute only for every morphism witnessing $f\colon N_{x'} = N_x$ (by definition of the morphisms in $\mathbf{N}^{=}_{\mathcal{X}}$), i.e., whenever $N_{x'} = N_x$. Again, we have already argued that this is exactly the neighbourhood-uniformity condition.

In the remainder of this thesis, we will denote the subcategory of $\mathbf{CCA}_{\mathsf{G}}$ of coalgebras where γ is a natural transformation $Q^{=} \Rightarrow \Delta_S$ (neighbourhood-uniform) and $Q \Rightarrow \Delta_S$ (extension-uniform) by $\mathbf{CCA}_{\mathsf{NU}}$ and $\mathbf{CCA}_{\mathsf{EU}}$ respectively.

4.4 Summary of general CAs

This section defined the class of *general CA* (Definition 4.6): a generalisation of CA (in comparison to Definition 2.13) that also allows non-uniform CA and CA without root. We constructed a functor C_{G} and a category $\mathbf{CCA}_{\mathsf{G}}$ of C_{G}-coalgebras that are exactly the general CA.

We also defined two subclasses of general CA, namely the *neighbourhood-uniform CA* and the *extension-uniform CA*. These classes consist of CA in which different cells may have different local rules, but where CA with similar neighbourhood structures must have related local rules. We showed that these arise coalgebraically as the subcategories $\mathbf{CCA}_{\mathsf{NU}}$ and $\mathbf{CCA}_{\mathsf{EU}}$, respectively, of coalgebras in $\mathbf{CCA}_{\mathsf{G}}$ whose local rule satisfies a naturality condition.

In the remainder of this thesis we will investigate the options of defining a modal logic over coalgebraic CA. We will take the coalgebras in the category $\mathbf{CCA}_{\mathsf{G}}$ as our working definition of "CA".

5 Coalgebra homomorphisms and behavioural equivalence

An important theoretical tool that can automatically construct modal logic languages for coalgebras is the *predicate lifting* framework developed by Pattinson [30, 31] and Schröder [41] for coalgebras of endofunctors on **Set**. See Klin [22] for a treatment of base categories beyond **Set**. This framework uses *behavioural equivalence* (BE, Definition 2.21) as the main notion of similarity. In particular, the language that this framework provides ensures that cells $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ satisfy the same formulas iff $x \overset{\mathsf{BE}}{\approx} y$. A consequence that any differences between cells $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ cannot be described by the logic as soon as $x \overset{\mathsf{BE}}{\approx} y$. We will see that this unfortunately means that the framework is inappropriate to describe spacial-temporal patterns in trace sequences of CA, since certain pairs of CA with very different dynamics turn out to be behaviourally equivalent. Phrased differently: to fulfill our research question, a model logic must apparently be able to distinguish behaviourally equivalent coalgebras.

Before we can justify this conclusion, we need to develop a clear overview of what behavioural equivalence means in context of CA. To this end, we will first analyse homomorphisms between CA. This will reveal that coalgebra homomorphisms are characterised two properties: (1) they commute with cowriter-comonad coalgebras and (2) they witness an extension condition similar to the one of the class of extension-uniform CA (Definition 4.7). Thereafter we construct two methods of characterising behaviourally equivalent cells: (1) contracting a CA such that it has no distinct behaviourally equivalent cells anymore, and (2) constructing a terminal coalgebra in $\mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ that has exactly one cell for every equivalence class of behaviourally equivalent cells in $\mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ (for a any signature Σ).

Throughout this section, we assume a fixed signature $\Sigma = (\mathscr{M}, N, S)$. Analogously to $\mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ (Definition 4.16), we define $\mathbf{CCA}^{\Sigma}_{\mathsf{NU}}$ to be the full subcategory of $\mathbf{CCA}_{\mathsf{NU}}$ of coalgebras defined of Σ (equivalently, it is the full subcategory of $\mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ of neighbourhood-uniform CA).

5.1 Characterising coalgebra homomorphisms

We will begin the investigation of behavioural equivalence by characterising the properties of coalgebra homomorphisms in $\mathbf{CCA}_{\mathsf{G}}^{\Sigma}$. These homomorphisms have two properties, because the output of the object map of the functor C_{G} (Definition 4.12) is a Cartesian product between two sets. The first property states that coalgebra homomorphisms commute with the cowriter coalgebras, and the second property states an extension relation between local rules of cells in the domain and image, similar to the extension relation between local rules in extension-uniform CA (Definition 4.8).

In this subsection, we assume two fixed general CA $\mathcal{X} = (X, \langle \chi, \gamma \rangle), \mathcal{Y} = (Y, \langle \xi, \delta \rangle) \in \mathbf{CCA}_{\mathsf{G}}^{\Sigma}$.

Both properties follow from the coalgebra homomorphism law (Eq. (6)). For a coalgebra homomorphism $h \colon \mathcal{X} \to \mathcal{Y}$ this law asserts that the following diagram commutes:

$$\begin{array}{ccc} X & \xrightarrow{\quad h \quad} & Y \\ {\scriptstyle \langle \chi, \gamma \rangle}\downarrow & & \downarrow{\scriptstyle \langle \xi, \delta \rangle} \\ \sum\limits_{f \in X^{\mathscr{M}}} S^{S^{\mathsf{Prelm}_N(f)}} & \xrightarrow[C_{\mathsf{G}} h]{} & \sum\limits_{f' \in Y^{\mathscr{M}}} S^{S^{\mathsf{Prelm}_N(f')}} \end{array} \tag{34}$$

Both paths from X (upper left) to $\sum\limits_{f' \in Y^{\mathscr{M}}} S^{S^{\mathsf{Prelm}_N(f')}}$ (lower right) in Diagram (34) output a tuple of the form $((-) \mathbin{\textcircled{\xi}} y, \delta_y)$ on any input $x \in X$. The diagram commutes for evert coalgebra homomorphism h, which means that both the upper-left path and the right-lower path output the same tuple. The upper-left path is the morphism:

$$\langle \chi, \delta \rangle \circ h$$

and the right-lower path is:

$$C_{\mathsf{G}} h \circ \langle \chi, \gamma \rangle = (h \circ ((-) \mathbin{\textcircled{\chi}} \chi), \lambda x \,.\, \lambda k \,.\, \gamma_x(\lambda \, |n|_x \,.\, k(|n|_{h(x)})$$

by definition of C_{G} (Definition 4.12), where the type of the bound variable k is $N_{h(x)} \to S$.

Remark 5.1. The discussion above used the notational simplifications indicated in Remark 4.14. We wrote N_x for $\mathsf{Prelm}_N((-) \mathbin{\textcircled{\chi}} x) = \mathsf{Prelm}_N(\chi(x))$, $N_{h(x)}$ for $\mathsf{Prelm}_N(\xi(h(x)))$ and $|n|_x$ for $|n|_{\chi(x)}$. Finally, we also used the following simplification:

$$\begin{aligned} |n|_{h(x)} &= |n|_{(-) \mathbin{\textcircled{\xi}} h(x)} \\ &= |n|_{\xi(h(x))} \\ &= |n|_{(\xi \circ h)(x)} \,. \end{aligned}$$

5.1.1 First property: commutativity with cowriter-comonad coalgebras

We first consider the equality of the first elements of the tuples that the paths in Diagram (34) output. This equality asserts that h is a $\mathscr{M}$-χ-ξ-homomorphism, defined as follows:

Definition 5.2: $\mathscr{M}$-χ-ξ-homomorphism

A $\mathscr{M}$-χ-ξ-homomorphism between the cowriter-comonad coalgebras (X, χ) and (Y, ξ) is a function $h \colon X \to Y$ with the following property: for all $x \in X$ and $m \in \mathscr{M}$, it holds that

$$h(m \mathbin{\textcircled{\chi}} x) = m \mathbin{\textcircled{\xi}} h(x). \tag{35}$$

A M-χ-ξ-homomorphism h is an *$\mathscr{M}$-restricted surjection* (*injection*/*isomorphism* respectively) if, for

every $x \in X$, the restriction

$$h\restriction(\mathscr{M} \mathbin{⦶} x)\colon (\mathscr{M} \mathbin{⦶} x) \to (\mathscr{M} \mathbin{⦷} h(x)) \tag{36}$$

is a surjection (injection/bijection respectively).

Lemma 5.3

Every coalgebra homomorphism $h\colon \mathcal{X} \to \mathcal{Y}$ is a $\mathscr{M}$-χ-ξ-homomorphism.

Proof. The first projections of the composite paths in Diagram (34) gives $\xi \circ h = (h \circ (-)) \circ \chi$, i.e.,

$$\xi \circ h = \lambda x \,.\, \lambda m \,.\, h(\chi(x)(m)).$$

On inputs $x \in X$ and $m \in \mathscr{M}$ this evaluates, in infix notation, to:

$$m \mathbin{⦷} h(x) = h(m \mathbin{⦶} x).$$

□

An important property of M-χ-ξ-homomorphisms is that they are always *$\mathscr{M}$-restricted surjective.*

Lemma 5.4

Every M-χ-ξ-homomorphism $h\colon (X, \chi) \to (Y, \xi)$ is an $\mathscr{M}$-restricted surjection.

Proof. Take any $x \in X$, then we need to show that $\mathscr{M} \mathbin{⦷} h(x)$ is contained in the image of h under $\mathscr{M} \mathbin{⦶} x$. To this end, take an arbitrary $m \in M$, the property of an M-χ-ξ-homomorphism asserts that:

$$h(m \mathbin{⦶} x) = m \mathbin{⦷} h(x),$$

and indeed $(m \mathbin{⦶} x) \in (\mathscr{M} \mathbin{⦶} x)$. □

Remark 5.5. It might still be possible that there exist an $y \in Y$ for which there exist no $m \in \mathscr{M}$ such that $y = m \mathbin{⦷} h(x)$. In this case y may not be in the image of h under $\mathscr{M} \mathbin{⦶} x$.

We provide two examples of coalgebra homomorphisms: the first one is $\mathscr{M}$-restricted injective, the second not.

Example 5.6 (Coalgebra isomorphism)**.** Recall Example 4.10, which showed that we can we can construct a neighbourhood-uniform CA on the forktree monoid $\mathscr{F}$ whose graph structure is the same as the forktree. Assume such a neighbourhood-uniform CA $\mathcal{X} \in \mathbf{CCA}^{\Sigma}_{\mathsf{NU}}$, and we will construct a coalgebra homomorphism $\mathcal{X} \to \mathcal{X}$. Let h be the the mapping $h\colon \mathscr{F} \to \mathscr{F}$ that appends the suffix LL to every string, that is,

$$h\colon x \mapsto xLL.$$

It sends the root ε to LL, but the subtree rooted in LL is isomorphic to $\mathscr{F}$ (as seen in Figure 5). Furthermore, it preserves the structure of the forktree: it sends cells to cells with the same neighbourhood. For example, h sends the children of the root L and R to LLL and RLL, respectively, and the only child of L, which is $LR = LL$ to the only child of LLL, which is $LLLL = LRLL$, etc. Therefore h is a M-χ-χ-homomorphism, the first property of a coalgebra homomorphism by Theorem 5.9.

Neighbourhood-uniformity ensures that cells with the same neighbourhood have the same local rule, which implies that the extension property is satisfied. Thus from Theorem 5.9 it follows that h is indeed a coalgebra homomorphism.

Example 5.7 (Non-injective coalgebra homomorphism)**.** Example 4.11 showed that we can we can construct an extension-uniform CA on the three-forktree monoid $3\mathscr{F}$ whose graph structure is exactly the three-forktree. Let $\mathcal{X} \in \mathbf{CCA}_{\mathsf{EU}}$ be such an extension-uniform CA. We will construct a non-injective coalgebra homomorphism $h\colon \mathcal{X} \to \mathcal{X}$.

Recall that all cells whose string starts with the character M have the same neighbourhood N_M, and therefore the same local rule γ_M. Furthermore, we saw that the local rules of all other cells need to extend γ_M. Therefore the extension-property of Theorem 5.9 is always satisfied if we map cells to cells whose strings starts with the character M. Thus we conjecture that we can define h as follows: it sends every string $s \in 3\mathscr{F}$ to Ms' where s' is s but with all occurrences of L and R replaced by M. For example, $h(MRR) = MMMM$, $h(\varepsilon) = M$ and $h(MLRML) = MMMMMM$. Note that this mapping is independent of the choice of representative of equivalence classes under congruence, since by definition of $3\mathscr{F}$, all generating equalities only equate strings of the same length (see Example 4.11, it can also be seen in Figure 8). Put visually, h maps every cell of depth k in the tree to the cell M^{k+1} of depth $k+1$ in the central branch of Figure 8.

It follows that for all monoid elements $m \in 3\mathscr{F}$ and every cell $x \in X = 3\mathscr{F}$ that $h(m \mathbin{⨂} x) = M^{k_m + k_x}$ where k_m and k_x are the lengths of the strings m and x respectively. But we can write $m = s_0 s_1 s_2 \dots s_{k_m}$ for $s_i \in \{L, M, R\}$ and $\ell \in \mathbb{N}$. Then the generating equalities of $3\mathscr{F}$ and the cowriter-comonad property of $⨂$ (Eq. (12)) ensure that:

$$
\begin{aligned}
m \mathbin{⨂} x &= s_{k_m} \mathbin{⨂} s_{k_m - 1} \mathbin{⨂} \dots \mathbin{⨂} s_1 \mathbin{⨂} s_0 \mathbin{⨂} h(x) \\
&= s_{k_m} \mathbin{⨂} s_{k_m - 1} \mathbin{⨂} \dots \mathbin{⨂} s_1 \mathbin{⨂} s_0 \mathbin{⨂} M^{k_x} \\
&= M^{k_x} s_0 s_1 \dots s_{k_m - 1} s_{k_m} \\
&= M^{k_x} M s_1 \dots s_{k_m - 1} s_{k_m} && \text{// Generating equalities of } 3\mathscr{F}. \\
&= \dots \\
&= M^{k_x} M^{k_m - 1} s_{k_m} && \text{// Generating equalities of } 3\mathscr{F}. \\
&= M^{k_x} M^{k_m} && \text{// Generating equalities of } 3\mathscr{F}. \\
&= M^{k_x + k_m} \\
&= h(m \mathbin{⨂} x),
\end{aligned}
$$

which shows that h is also a M-χ-χ-homomorphism. Thus h has all properties of a coalgebra homomorphism as required by Theorem 5.9. Now h is not injective, as it collapses the three-forktree into a tree with branching factor 1 (e.g., it maps L and R, which are distinct cells, both to MM).

5.1.2 Second property: extension of local rules

Now we analyse the implication of equality of the second elements of the tuples that the paths in Diagram (34) output. It asserts that the local rule of a cell in the image of a coalgebra homomorphism extends (Definition 4.7) the local rule of the corresponding cell in the domain.

Lemma 5.8

A $\mathscr{M}$-χ-ξ-homomorphism $h\colon (X, \chi) \to (Y, \xi)$ is a coalgebra homomorphism $h\colon \mathcal{X} \to \mathcal{Y}$ if and only if it has the property that γ_x extends $\delta_{h(x)}$ for all $x \in X$.

Proof. Unfolding the definition of an extension (Definition 4.7), we need to show that for all $x \in X$ and all $N_{h(x)}$-compatible $z\colon N \to S$ (Definition 4.2) it holds that:

$$\delta_{h(x)}(z \setminus\setminus h(x)) = \gamma_x(z \setminus\setminus x).$$

The second projection of the output of the paths in Diagram 34 asserts that for all $x \in X$ and $k \in S^{N_{h(x)}}$ it holds that

$$(\delta \circ h)(x)(k) = (C_{\mathsf{G}} h \circ \gamma)(x)(k) \tag{37}$$

But note that $N_{h(x)}$-compatible elements of S^N are in bijection with $S^{N_{h(x)}}$ (with the bijection given by $(-)\!\!\setminus\!\!h(x)$). Thus the above statement is equivalent to the statement that for every $N_{h(x)}$-compatible $z\colon N \to S$ it holds that:

$$(\delta \circ h)(x)(z\!\!\setminus\!\!h(x)) = (C_{\mathsf{G}} h \circ \gamma)(x)(z\!\!\setminus\!\!h(x)).$$

Simplifying both sides gives

$$\begin{aligned}
\delta_{h(x)}(z\!\!\setminus\!\!h(x)) &= (C_{\mathsf{G}} h(\gamma_x))(z\!\!\setminus\!\!h(x)) \\
&= \gamma_x(\lambda n_x \,.\, (z\!\!\setminus\!\!h(x))(|n|_{h(x)}) && \text{// Definition } C_{\mathsf{G}}h \text{ (Definition 4.12).} \\
&= \gamma_x(\lambda n_x \,.\, z(n)) && \text{// Definition } \setminus\!\!h(x) \text{ (Definition 4.4).} \\
&= \gamma_x(\lambda n_x \,.\, (z\!\!\setminus\!\!x)(|n|_x) \qquad = \gamma_x(z\!\!\setminus\!\!x), && \text{// Definition } \setminus\!\!x \text{ (Definition 4.4).}
\end{aligned}$$

as required. □

We can summarise the characterisation of coalgebra homomorphisms as follows: convenient notation as follows:

Theorem 5.9: Characterisation of $\mathbf{CCA}^{\Sigma}_{\mathsf{G}}$-morphisms

The coalgebra homomorphisms $h\colon \mathcal{X} \to \mathcal{Y}$ in $\mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ are exactly the M-χ-ξ-homomorphism with the following extension property: extension property: for all $x \in X$ and $N_{h(x)}$-compatible $z\colon N \to S$ it holds that:

$$\delta_{h(x)}(z\!\!\setminus\!\!h(x)) = \gamma_x(z\!\!\setminus\!\!x). \tag{38}$$

5.2 Illustration of behavioural equivalence

Theorem 5.9 characterises the coalgebra homomorphisms, but on a very abstract level. We will now show several examples and more specific results to obtain an intuition when cells are behaviourally equivalent.

We begin with a basic example of finding a coalgebra homomorphism (in this case even an isomorphism) via Theorem 5.9:

Example 5.10. Define the data:

$$\begin{aligned}
\mathscr{M} &= \mathbb{Z} \\
N &= \{-1, 0, 1\} \\
\mathcal{X} &= \{\mathbb{Z}^+ = \{m^+\}_{m\in\mathbb{Z}}, \langle \chi, \gamma^+ \rangle\} \\
&\qquad \text{with } n \mathbin{⨀} m \stackrel{\text{def}}{=} n + m \\
\mathcal{Z} &= \{\mathbb{Z}^- = \{m^-\}_{m\in\mathbb{Z}}, \langle \xi, \gamma^- \rangle\} \\
&\qquad \text{with } n \mathbin{⨁} m \stackrel{\text{def}}{=} m - n
\end{aligned}$$

where $\gamma^+_{n^+} = \gamma^-_{n^-}$ is an arbitrary local rule on an arbitrary state set S. Note that all cells have $N_{n^+} = N_{n^-} \cong N$, so with a minor abuse of notation we will write all neighbourhood quotients as N.

Now define $h\colon \mathbb{Z}^+ \to \mathbb{Z}^-$ as negation: $h(n) \stackrel{\text{def}}{=} -n$. We can verify that h is a coalgebra homomorphism by checking it has the two required properties (see Theorem 5.9).

To check that h is a $\mathscr{M}$-χ-ξ-homomorphism, observe that:

$$h(n \mathbin{⊛} m) = h(n+m) = -(n+m) = -n - m = (-m) - n = n \mathbin{◎} h(m).$$

To check that h satisfies the extension relation Eq. (38), take any $z \colon N \to S$, observe that:

$$(z \lVert h(n^+))(|m|_{h(n^+)}) = z(m) = (z \lVert n^+)(|m|_{n^+})$$

thus when $\gamma^-_{h(n^+)}$ queries $m \in N$ to the input configuration $z \lVert h(n^+)$, it obtains the same state in S as when $\gamma^+_{n^+}$ queries m to its own input $z \lVert n^+$. Since $\gamma^+_{n^+} = \gamma^-_{n^-}$, they must therefore give the same outputs on those inputs.

Note that the m-neighbour of $h(n^+)$ is $m \mathbin{◎} -n^- = -n^- - m$ whereas the m-neighbour of n^+ is $n^+ + m$. This does not matter for the extension relation, which only depends on whether paths in N lead to distinct cells.

Lemma 5.11

Every cell in a uniform CA $\mathcal{X} = (X, \langle \chi, \gamma \rangle) \in \mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ (i.e., $\gamma_x = \gamma_{x'}$ for all $x, x' \in X$) is behavioural equivalent to the cell of a single-celled CA.

Proof. Let $\mathcal{X} = (X, \langle \chi, \gamma \rangle)$ be a uniform CA on the signature $(\mathscr{M}, N, S)$. If $X = \emptyset$ the lemma holds vacuously, so assume that X is not empty.

We construct a single-celled CA $\mathcal{U} = (\{*\}, \langle \xi, \delta \rangle) \in \mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ on the same signature. The cowriter-comonad coalgebra of $\mathcal{U}$ is trivial: $(-) \mathbin{◎} * \stackrel{\text{def}}{=} \lambda m \,.\, *$.

As for the local rule δ, first note that the neighbourhood quotient $N_* = \{N\}$ is a singleton set, so all neighbourhood configurations $z \colon N_* \to S$ are of the form $\mathsf{const}_s = \lambda\, |n|_* \,.\, s \colon N_* \to S$ for some $s \in S$. Since $\mathcal{U}$ has only one cell, δ has only one component $\delta_* \colon N_* \to S$ that we need to define. Since X is not empty, there exists an $x \in X$, and uniformity asserts that γ_x is the same for any choice of x. Therefore we can define δ_* unambiguously as follows:

$$\delta_*(\mathsf{const}_s) \stackrel{\text{def}}{=} \gamma_x(\lambda\, |x| \,.\, s).$$

Now $h \colon \mathcal{X} \to \mathcal{U}$ defined as $h = \mathsf{const}_*$ satisfies all properties of a coalgebra homomorphism (Theorem 5.9), and indeed sends all cells of $\mathcal{X}$ to a CA with a single cell. □

Example 5.12 (CGOL is behavioural equivalent to a constant dead CA)**.** The proof of Lemma 5.11 shows that two cells $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ of two uniform CA are behaviourally equivalent as soon as γ_x and δ_y agree on constant neighbourhood configurations (inputs in which all neighbours have the same state).

For CGOL (Examples 2.17 and 3.6) this means that all cells are behavioural equivalent to a rather trivial single-celled CA whose single cell always becomes ∘ the next timestep. Indeed, we observe that CGOL's local rule (Eq. (5)) always outputs ∘ on constant neighbourhood configurations $z \colon N \to \{\bullet, \circ\}$ (for $z = \mathsf{const}_\circ$, the target cell and all neighbours are dead, so it stays dead. For $z = \mathsf{const}_\bullet$, the cell has too many alive neighbours to stay alive).

This observation is quite significant, since CGOL is universal (both Turing universal, as well as universal for 2D CA [9]), while the dynamics of the constant-dead-single-cell CA are completely trivial.

5.3 The terminal coalgebra

With a practical characterisation of coalgebra homomorphisms at hand (theorem 5.9), we can construct a more complete overview of the behavioural equivalence relation. The terminal object of $\mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ provides a succinct characterisation of behavioural equivalent cells: all pairs of behavioural equivalent cells are mapped to the same cell in this terminal coalgebra. In this subsection we construct this terminal coalgebra.

5.3.1 Coalgebra contractions

We will construct the terminal coalgebra by (1) *contracting* every CA $\mathcal{X} \in \mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ into a canonical form $\|\mathcal{X}\| \in \mathbf{CCA}^{\Sigma}_{\mathsf{G}}$, such that there exist exactly one $\iota_X \colon \mathcal{X} \to \|\mathcal{X}\|$, and (2) combining all contractions in one coalgebra. The second step requires some additional care to avoid quantifying over a large collection that is not a set.

Definition 5.13: Contraction

Given a coalgebra $\mathcal{X} = (X, \langle \chi, \gamma \rangle) \in \mathbf{CCA}^{\Sigma}_{\mathsf{G}}$, define the *contraction* $\|\mathcal{X}\|$ of $\mathcal{X}$ to be the C_{G}-coalgebra $(\|X\|, \langle \|\chi\|, \|\gamma\| \rangle)$ where:

- $\|X\|$ are the equivalence classes $\|x\|$ (with $x \in X$) under the transitive closure of $\overset{\mathsf{BE}}{\approx}$.
- $m \mathbin{\|\circledS\|} \|x\| \overset{\text{def}}{=} \|m \circledS x\|$.
- $\|\gamma\|_{\|x\|} \colon S^{N_{\|x\|}} \to S$ is defined to be $\gamma_x \circ u$ where

$$u \colon S^{N_{\|x\|}} \to S^{N_x}$$
$$u(k)(|n|_x) \overset{\text{def}}{=} k(|n|_{\|x\|}),$$

  which is well defined only if $N_{\|x\|} \leq N_x$ for all $x \in \|x\|$. This indeed holds; see Theorem 5.14 below.

Lemma 5.18 will show that $\|\mathcal{X}\| \in \mathbf{CCA}^{\Sigma}_{\mathsf{G}}$.

Theorem 5.14

For all $x \in X$ it holds that:

$$N_{\|x\|} \leq \bigwedge_{x \in \|x\|} N_x \leq N_x.$$

This theorem is easiest to prove with the help of a new definition and two lemmas. For convenience, denote $N_\wedge \overset{\text{def}}{=} \bigwedge_{x \in \|x\|} N_x$ and the classes in $N_\wedge$ by $|n|_\wedge$.

Definition 5.15: Domino relation

Let $U \in \mathbf{Set}$ and let $\{A_i \subseteq U\}_{i \in I}$ be a set of subsets of U. Then there exists a *domino path* from $a \in U$ to $b \in U$ if there exists an $n \in \mathbb{N}$ and a sequence $(x_i, A_i)_{1 \leq i \leq n+1}$ such that $a \in A_0$, $b \in A_{n+1}$ (x_{n+1} may differ from b) and

$$x_{i+1} \in A_i \cap A_{i+1} \qquad \text{for all } 0 \leq i \leq n.$$

This relation is an equivalence relation $\boxplus \subseteq U \times U$:

- The singleton sequence $((a, A))$ proves the reflexive case.
- Symmetry follows from the sequence (x_{n+1-i}, A_{n+1-i}), which gives a domino path from b to a.
- Transitivity is simply concatenating sequences.

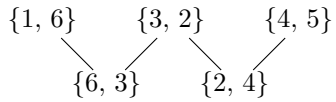


Figure 10: Example of a domino path proving $1 \boxplus 5$ in the collection $U = \{\{1,6\},\{6,3\},\{3,2\},\{2,4\},\{4,5\}\}$. Note that we can interpret the elements of U as domino pieces in the actual game, thus justifying the name of the relation.

Lemma 5.16

Consider the domino relation over the N-partitions $\{N_x \mid x \in \|x\|\}$, then

$$|m|_{\wedge} = |m'|_{\wedge} \text{ iff } m \boxplus m'. \tag{39}$$

Proof. We show both directions, to begin with right-to-left. Let $(n_i, |n_i|_{\wedge})_{1 \leq i \leq k}$ be the domino sequence proving $m \boxplus m'$. All sets in $N_{\wedge}$ are mutually disjoined, but $n_{i+1} \in |n_i|_{\wedge}$ and $n_{i+1} \in |n_{i+1}|_{\wedge}$. Therefore it must be that $|n_i|_{\wedge} = |n_{i+1}|_{\wedge}$, which implies all sets in the sequence are equal, and in particular $|m|_{\wedge} = |m'|_{\wedge}$.

We prove the other direction via contraposition: assume that there exists no domino path from m to m', but still $|m|_{\wedge} = |m'|_{\wedge}$. We will derive a contradiction by constructing an $N_{\Delta} \in \mathsf{PartOrd}(N)$ such that $N_{\wedge} \lneq N_{\Delta} \leq N_{x'}$ for all $x' \in \|x\|$, which cannot exist since $N_{\wedge}$ is the greatest lower bound.

Let $N_{\Delta} = \{|n|_{\Delta}\}_{n \in N}$ where:

$$|n|_{\Delta} \stackrel{\text{def}}{=} \{n' \in N \mid n \boxplus n'\}.$$

Since we are using domino paths over $\{N_{x'} \mid x' \in \|x'\|\}$ it holds that $\bigcup_{x' \in \|x'\|} |m|_{x'} \subseteq |m|_{\Delta}$ (since clearly $m \boxplus n$ for all $n \in |m|_{x'}$ via a length-1 path, so $|m|_{x'} \subseteq |m|_{\Delta}$). Hence $N_{\Delta} \leq N_x$.

It remains to show that $N_{\wedge} \lneq N_{\Delta}$. We first show $N_{\wedge} \leq n_{\Delta}$. Take any $n \in N$, then it is to show that $|n|_{\Delta} \subseteq |n|_{\wedge}$. For any $n' \in N$ we have $n \boxplus n'$ iff it holds that $|n|_{\Delta} = |n'|_{\Delta}$. But we have already shown the right-to-left direction of the current lemma, thus we also know that $n \boxplus n'$ implies $|n|_{\wedge} = |n'|_{\wedge}$ Hence $|n|_{\Delta} = |n'|_{\Delta}$ implies $|n|_{\wedge} = |n'|_{\wedge}$, i.e., $|N|_{\Delta} \subseteq |n|_{\wedge}$, as required.

The last step is to show that $N_{\wedge} \neq N_{\Delta}$. By our assumption, $m \not\boxplus m'$, and thus $|m|_{\Delta} \neq |m'|_{\Delta}$, while we assumed $|m|_{\wedge} = |m'|_{\wedge}$. Thus we conclude $N_{\wedge} \lneq N_{\Delta} \leq N_x$ for all $x \in \|x\|$, which is our sought contradiction. □

Lemma 5.17

For $x \in X$ and $y \in Y$ (given coalgebras $\mathcal{X} = (X, \langle \chi, \gamma \rangle)$ and $\mathcal{Y} = (Y, \langle \xi, \delta \rangle)$), if $x \overset{\mathsf{BE}}{\approx} y$ then $(m \circledast x) \overset{\mathsf{BE}}{\approx} (m \circledast y)$.

Proof. If $x \overset{\mathsf{BE}}{\approx} y$ then there exists a $\mathcal{Z} = (Z, \langle \zeta, \alpha \rangle)$ and $h \colon \mathcal{X} \to \mathcal{Z}$ and a $k \colon \mathcal{Y} \to \mathcal{Z}$ such that $h(x) = k(y)$. From the $\mathscr{M}$-χ-ζ- and $\mathscr{M}$-ξ-ζ-homomorphism properties of h and k respectively it follows that

$$h(m \circledast x) = m \circledcirc h(x) = m \circledcirc k(y) = k(m \circledast y),$$

which indeed proves $(m \circledast x) \overset{\mathsf{BE}}{\approx} (m \circledast y)$. □

We can now prove Theorem 5.14:

Proof. For convenience, denote the classes $N_{\|x\|} = \{|n|_\bullet \mid n \in N\}$, and take any $n \in N$; then it is to show that $|n|_\wedge \subseteq |n|_\bullet$. To this end, we pick arbitrary $m, m' \in |n|_\wedge$ and show that $|m|_\bullet = |m'|_\bullet$. (If we do this for all $m \in |n|_\wedge$, then it follows that $|n|_\bullet = |m|_\bullet$ for all such m, so $m \in |n|_\bullet$ and it follows that indeed $|n|_\wedge \subseteq |m|_\bullet$). There either exists or not exists an $x_0 \in \|x\|$ such that $y := m \circledast x_0 = m' \circledast x_0 =: y'$. If such an x_0 exists, then the identity coalgebra homomorphism shows $y \overset{\mathsf{BE}}{\approx} y'$, which implies $\|y\| = \|y'\|$. Hence both m and m' have that

$$m \,\|\circledast\|\, \|x_0\| = \|m \circledast x_0\| = \|y\| = \|y'\| = \|m' \circledast x_0\| = m' \,\|\circledast\|\, \|x_0\|,$$

which proves that m and m' lead from $\|x_0\|$ to the same cell $\|y\|$ via $\|\circledast\|$, thus $|m|_\bullet = |m'|_\bullet$ by definition of neighbourhood quotients.

The case where no such x_0 exists remains. From Lemma 5.16 it still follows that $m \boxplus m'$. So we obtain a domino sequence $(n_i, |n_i|_{x_i})_{1 \le i \le k}$ such that $x_i \in \|x\|$, $n_{i+1} \in |n_i|_{x_i} \cap |n_{i+1}|_{x_{i+1}}$, $m = n_0$ and $m' \in |n_{k+1}|_{x_{k+1}}$. From the definition of neighbourhood quotients and the fact that $n_{i+1} \in |n_i|_{x_i}$ it follows that

$$n_i \circledast x_i = n_{i+i} \circledast x_i.$$

Therefore the identity coalgebra homomorphism proves that $n_i \circledast x_i \overset{\mathsf{BE}}{\approx} n_{i+1} \circledast x_i$, so by definition of $\|(-)\|$ we find $\|n_i \circledast x_i\| = \|n_{i+1} \circledast x_i\|$. But by definition of $\|\circledast\|$ this means that $n_i \,\|\circledast\|\, \|x_i\| = n_{i+1} \,\|\circledast\|\, \|x_i\|$. Thus from the definition of neighbourhood quotients $|(-)|_\bullet = |(-)|_{\|x\|}$ (now in the contracted coalgebra) it follows that $|n_i|_\bullet = |n_{i+1}|_\bullet$. This holds for all $0 \le i \le k$, thus by transitivity we find the desired $|m|_\bullet = |m'|_\bullet$. □

The following lemma shows that the output of a contraction is again in $\mathbf{CCA}^\Sigma_{\mathsf{G}}$, i.e., the first component is again a cowriter-comonad coalgebra:

Lemma 5.18

If $\mathcal{X} \in \mathbf{CCA}^\Sigma_{\mathsf{G}}$ then $\|\mathcal{X}\| \in \mathbf{CCA}^\Sigma_{\mathsf{G}}$. That is, $\|\circledast\|$ is a cowriter-comonad coalgebra that satisfies Eq. (11) and (12).

Proof. Take any $n, m \in \mathscr{M}$ and $\|x\| \in \|X\|$ (for some $x \in X$), then we verify that Eq. (11) and (12) are satisfied as follows:

$$\mathbf{1} \,\|\circledast\|\, \|x\| \overset{\text{def}}{=} \|\mathbf{1} \circledast x\| = \|x\|$$
$$n \,\|\circledast\|\, (m \,\|\circledast\|\, \|x\|) \overset{\text{def}}{=} n \,\|\circledast\|\, \|m \circledast x\| \overset{\text{def}}{=} \|(n \bullet m) \circledast x\| \overset{\text{def}}{=} (n \bullet m) \,\|\circledast\|\, \|x\|.$$

□

Example 5.19 (Example of a contraction)**.** Figure 11 visualises the effect of a contraction of a general CA. We assume that $\gamma_a = \gamma_{a'}$ and $\gamma_b = \gamma_{b'}$, which implies that $a \overset{\mathsf{BE}}{\approx} a'$ and $b \overset{\mathsf{BE}}{\approx} b'$. We also assume that γ_a and γ_b disagree on constant configurations, which ensures that $a \overset{\mathsf{BE}}{\not\approx} b$ and $a \overset{\mathsf{BE}}{\not\approx} b$. If they would happen to agree on constant configurations, then the CA would be contracted to a single cell.

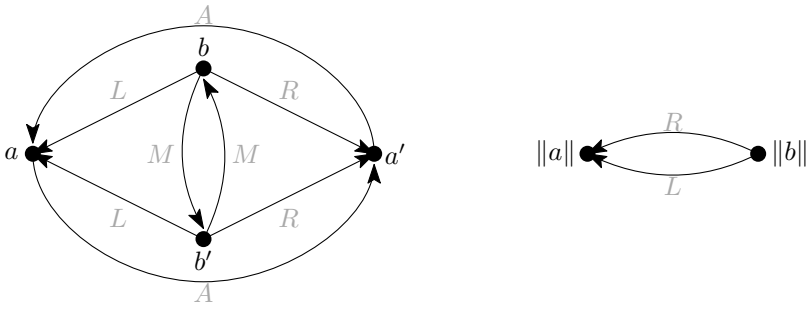


Figure 11: **Left:** a small CA on the monoid freely generated over $\{L, M, R, A\}$ with the empty string ε as identity element, neighbourhood $N = \{\varepsilon, L, M, R, A\}$ and four cells $X = \{a, a', b, b'\}$. Only paths in N are shown, with labels are indicated in gray; omitted paths (in N) implicitly denote paths to the cell itself. **Right:** contraction of the CA on the left, assuming that γ_a and γ_b disagree on constant configurations, but that $\gamma_a = \gamma_{a'}$ and $\gamma_b = \gamma_{b'}$.

Remark 5.20. It is possible that $N_{\|x\|} \lneq \bigwedge_{x' \in \|x\|} N_{x'}$. Take for example the neighbourhood-uniform CA $\mathcal{X} \in \mathbf{CCA}^{\Sigma}_{\mathsf{NU}}$ of Figure 12 below:

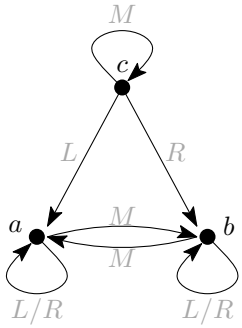


Figure 12: The neighbourhood-uniform CA $\mathcal{X} \in \mathbf{CCA}^{\Sigma}_{\mathsf{NU}}$ with $X = \{a, b, c\}$. The underlying monoid $\mathscr{M}$ is freely generated over the alphabet $\{L, M, R\}$ and has the empty string ε as identity element. The neighbourhood is $N = \{\varepsilon, L, M, R\}$. Only arrows in N are shown (but this is a generating set of $\mathscr{M}$), and the notation L/R means that both the paths L and R lead to the same neighbour.

The neighbourhood quotients of its cells are:

$$N_c = \{|\varepsilon, M|, |L, R|\}$$
$$N_a = N_b = \{|\varepsilon, L, R|, |M|\}.$$

These neighbourhoods are incompatible, thus γ_c can be chosen independently from γ_a and γ_b. Thus we can choose γ_c such that it disagrees with γ_a on constant inputs (e.g., pick $S = \{\bullet, \circ\}$, $\gamma_c = \mathsf{const}_\circ$ and $\gamma_a = \gamma_b = \mathsf{const}_\bullet$). In this case, c will not be contracted to a in $\|\mathcal{X}\|$, but a and b will both be contracted to $\|a\|$. We would obtain the contraction of Figure 13 below:

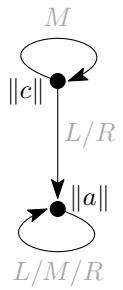


Figure 13: Contraction $\|\mathcal{X}\|$ of the CA $\mathcal{X}$ Figure 12 (in the case that the local rule at cell c disagrees with the local rule at cell a on constant configurations; otherwise the contraction would be a single cell).

In this contraction, it trivially holds that $\bigwedge_{x\in\|c\|} N_x = N_c$ (the meet of a singleton set), but $N_{\|x\|} = \{|\varepsilon, M|, |L, R|\}$. Thus $N_{|c|} \lneq \bigwedge_{x\in\|c\|} N_x$.

Before we can prove the uniqueness of $\iota_{\mathcal{X}}$, we need to show several other properties of contractions. But we can already start with defining the map itself:

Lemma 5.21

There exist a coalgebra homomorphism $\iota_{\mathcal{X}} \colon \mathcal{X} \to \|\mathcal{X}\|$.

Proof. The most obvious definition works:

$$\iota_{\mathcal{X}}(x) \stackrel{\text{def}}{=} \|x\| \tag{40}$$

for all $x \in X$. Applying the characterisation of coalgebra homomorphisms given by Theorem 5.9, we need to show that $\iota_{\mathcal{X}}$ (1) is a $\mathscr{M}$-χ-$\|\chi\|$-homomorphisms and (2) that $(z \mathbin{\backslash\!\backslash} x) = (z \mathbin{\backslash\!\backslash} \|x\|)$ for all $N_{\|x\|}$-compatible $z \colon N \to S$. This last requirement follows directly from the definition of $\|\gamma\|$ (Definition 5.13).

To show $\iota_{\mathcal{X}}$ is an $\mathscr{M}$-χ-$\|\chi\|$-homomorphism, simply observe that:

$$\iota_{\mathcal{X}}(m \circledast x) \stackrel{\text{def}}{=} \|m \circledast x\| = m \, \|\circledast\| \, \|x\| \stackrel{\text{def}}{=} m \, \|\circledast\| \, \iota_{\mathcal{X}}(x).$$

□

The next observation is rather trivial, but of great importance:

Lemma 5.22

There exist no distinct $u, v \in \|X\|$ such that $u \overset{\mathsf{BE}}{\approx} v$.

Proof. By definition of a contraction, we can write $u = \|x\|$ and $v = \|x'\|$ for some $x, x' \in X$. If there would exist $h \colon \|\mathcal{X}\| \to \mathcal{Y}$ and $\ell \colon \|\mathcal{X}\| \to \mathcal{Y}$ such that $h(u) = \ell(v)$, then $(h \circ \iota_{\mathcal{X}})(x) = (\ell \circ \iota_{\mathcal{X}})(x')$, which would imply $x \overset{\mathsf{BE}}{\approx} x'$. But then $u = \|x\| = \|x'\| = v$. □

Lemma 5.23

$\|(-)\|$ can be extended to an endofunctor $\mathbf{CCA}^{\Sigma}_{\mathsf{G}} \to \mathbf{CCA}^{\Sigma}_{\mathsf{G}}$.

Proof. Lemma 5.18 asserts that the object map of $\|(-)\|$ indeed outputs another object in $\mathbf{CCA}^{\Sigma}_{\mathsf{G}}$.
We define the action of $\|(-)\|$ on homomorphisms $h\colon \mathcal{X} \to \mathcal{Y}$ as follows:

$$\|h\|(\|x\|) \stackrel{\text{def}}{=} \|h(x)\| \tag{41}$$

for all $\|x\| \in \|x\|$.
Functoriality is immediate, but it is obvious not that $\|h\|$ is a coalgebra homomorphism; we prove this via the characterisation of coalgebra homomorphisms (Theorem 5.9). The first property we need to show is that $\|h\|$ is a $\mathscr{M}$-$\|\chi\|$-$\|\xi\|$-homomorphism. This can indeed be seen as follows (for arbitrary $\|x\| \in \|X\|$):

$$\begin{aligned}
&\|h\|(m \,\|\otimes\|\, \|x\|) \\
&\qquad = \|h\|(\|m \otimes x\|) && \text{// Definition } \|\otimes\| \text{.} \\
&\qquad = \|h(m \otimes x)\| && \text{// Definition } \|h\|\text{.} \\
&\qquad = \|m \circledast h(x)\| && \text{// } h \text{ is a } \mathscr{M}\text{-}\chi\text{-}\xi\text{-homomorphism.} \\
&\qquad = m \,\|\circledast\|\, \|h(x)\| && \text{// Definition } \|\circledast\| \text{.} \\
&\qquad = m \,\|\circledast\|\, \|h\|(\|x\|). && \text{// Definition } \|h\|\text{.}
\end{aligned}$$

The other required property states that, for all $N_{\|h(x)\|}$-compatible $z\colon N \to S$, it holds that:

$$\|\gamma\|_{\|x\|}(z \mathbin{\backslash\!\backslash} \|x\|) = \|\delta\|_{\|h(x)\|}(z \mathbin{\backslash\!\backslash} \|h(x)\|).$$

But we know h is a coalgebra homomorphism, so

$$\gamma_x(z' \mathbin{\backslash\!\backslash} x) = \delta_{h(x)}(z' \mathbin{\backslash\!\backslash} h(x)) \tag{42}$$

for all $N_{h(x)}$-compatible $z'\colon N \to S$. Since both $\iota_{\mathcal{Y}}$ and h are N-restricted surjections, it holds that $N_{\|H(x)\|} \leq N_{h(x)} \leq N_x$, so z is also $N_{h(x)}$- and N_x-compatible. Therefore:

$$\|\gamma\|_{\|x\|}(z \mathbin{\backslash\!\backslash} \|x\|) \stackrel{\text{def}}{=} \gamma_x(z \mathbin{\backslash\!\backslash} x) \stackrel{(42)}{=} \delta_{h(x)}(z \mathbin{\backslash\!\backslash} h(x)) \stackrel{\text{def}}{=} \|\delta\|_{\|h(x)\|}(z \mathbin{\backslash\!\backslash} \|h(x)\|).$$

□

Corollary 5.24

$\iota\colon \mathsf{Id}_{\mathsf{Coalg}(C_{\mathsf{G}})} \Rightarrow \|(-)\|$ is a natural transformation.

Proof. Directly from the definitions of $\iota_{\mathcal{X}}$ and $\|(-)\|$ on morphisms (see the proof of Lemma 5.23:

$$\|h\|(\iota_{\mathcal{X}}(x)) \stackrel{\text{def}}{=} \|h\|(\|x\|) \stackrel{\text{def}}{=} \|h(x)\| \stackrel{\text{def}}{=} \iota_{\mathcal{Y}}(h(x))$$

(for all $h\colon \mathcal{X} \to \mathcal{Y}$ and $x \in X$). Hence the naturality square commutes:

$$\begin{array}{ccc} \mathcal{X} & \xrightarrow{\;h\;} & \mathcal{Y} \\ \downarrow{\scriptstyle \iota_{\mathcal{X}}} & & \downarrow{\scriptstyle \iota_{\mathcal{Y}}} \\ \|\mathcal{X}\| & \xrightarrow{\;\|h\|\;} & \|\mathcal{Y}\| \end{array} \tag{43}$$

□

Lemma 5.25

For any coalgebra $\mathcal{X} \in \mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ it holds that $\|\mathcal{X}\| \cong \|\|\mathcal{X}\|\|$ with the isomorphism being $\iota_{\|\mathcal{X}\|}$.

Proof. No two distinct cells of $\|\mathcal{X}\|$ are behaviourally equivalent, so any coalgebra homomorphism $h\colon \|\mathcal{X}\| \to \|\|\mathcal{X}\|\|$ must be injective. Contracting never adds cells, thus h must also be surjective. Hence h must be an isomorphism; and $\iota_{\|\mathcal{X}\|}$ witnesses the existence of such a h. □

Theorem 5.26

The coalgebra homomorphism $\iota_{\mathcal{X}}\colon \mathcal{X} \to \|\mathcal{X}\|$ is the unique morphism between those coalgebras.

Proof. Suppose any other $h\colon \mathcal{X} \to \|\mathcal{X}\|$, and take any $x \in X$. Then there exists an $x' \in X$ such that $h(x) = \|x'\|$. From the proof of Lemma 5.23 it follows that the action of $\|h\|\colon \|\mathcal{X}\| \to \|\|\mathcal{X}\|\|$ maps $\|x\|$ to $\|h(x)\| = \|\|x'\|\|$ (recall that $\|\|X\|\|$ has as elements singleton sets $\|\|x\|\| \stackrel{\text{def}}{=} \{\|x\|\}$ for every $\|x\| \in \|X\|$. See the proof of Lemma 5.25).

From Lemma 5.25 it follows that $\iota_{\|\mathcal{X}\|}$ is an isomorphism, thus $\iota^{-1}_{\|\mathcal{X}\|} \circ \|h\|\colon \|\mathcal{X}\| \to \|\mathcal{X}\|$ is a coalgebra homomorphism that sends $\|x\|$ to $\|x'\|$. Thus $\|x\| \overset{\mathsf{BE}}{\approx} \|x'\|$ in $\|\mathcal{X}\|$; but a contraction has no distinct BE cells (Lemma 5.22), thus

$$\iota_{\mathcal{X}}(x) \stackrel{\text{def}}{=} \|x\| = \|x'\| = h(x).$$

Since x was arbitrary it follows that $\iota_{\mathcal{X}} = h$. □

It is important to realise that contractions may *not* preserve uniformity, neighbourhood-uniformity and extension-uniformity. Note that $\mathbf{CCA}^{\Sigma}_{\mathsf{NU}}$ includes all CA that have one of these three forms of uniformity, therefore we prove the following: if $\mathcal{X} \in \mathbf{CCA}^{\Sigma}_{\mathsf{NU}}$, then $\|\mathcal{X}\|$ might not be in $\mathbf{CCA}^{\Sigma}_{\mathsf{NU}}$.

Lemma 5.27

There exists a CA $\mathcal{X} \in \mathbf{CCA}^{\Sigma}_{\mathsf{NU}}$ such that $\|\mathcal{X}\| \notin \mathbf{CCA}^{\Sigma}_{\mathsf{NU}}$.

Proof. We construct a CA such that its contraction has two cells with the same neighbourhood quotients, but with different local rules. By definition of neighbourhood-uniformity (Definition 4.8, any CA in $\mathbf{CCA}^{\Sigma}_{\mathsf{NU}}$ has the property that all cells with the same neighbourhood have the same local rule.

Define the CA $\mathcal{X} = (X, \langle \chi, \gamma \rangle)$ with $X = \{x_i\}_{i\in\mathbb{Z}} \cup \{y_i\}_{i\in\mathbb{Z}}$, on the monoid $\mathscr{M}$ freely generated over the alphabet $\{L, M, R, A\}$ with the empty string ε as the unit element and neighbourhood $N = \{L, M, R, A\}$. We use the two-state alphabet $S = \{\bullet, \circ\}$, local rules $\gamma_{x_i} = \mathsf{const}_{\bullet}$ and $\gamma_{y_i} = \mathsf{const}_{\circ}$ for all $i \in \mathbb{Z}$, and cells connections χ defined as:

$$\begin{aligned} L \circledast x_i = M \circledast x_i &= x_{i-1} \\ R \circledast x_i &= x_{i+1} \\ A \circledast x_i &= y_i \\ L \circledast y_i &= y_{i-1} \\ M \circledast y_i = R \circledast y_i &= y_{i+1} \\ A \circledast y_i &= x_i, \end{aligned}$$

which is visualised in Figure 5.27.

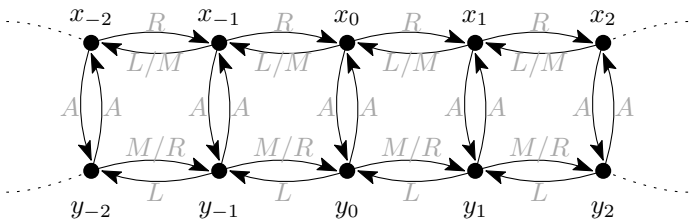


Figure 14: Fragments of the CA $\mathcal{X}$ of the proof of Lemma 5.27.

There exists a "shifting"-endohomomorphism $h_n \colon \mathcal{X} \to \mathcal{X}$ for every $n \in \mathbb{Z}$ defined as:

$$h_n(x_i) \stackrel{\text{def}}{=} x_{i+n}$$
$$h_n(y_i) \stackrel{\text{def}}{=} y_{i+n}$$

for all $i \in \mathbb{Z}$. Thus $x_i \overset{\mathsf{BE}}{\approx} x_{i+n}$ and $y_i \overset{\mathsf{BE}}{\approx} y_{i+n}$ for all $i, n \in \mathbb{Z}$. Note that $x_i \overset{\mathsf{BE}}{\not\approx} y_j$ for all $i, j \in \mathbb{Z}$ since γ_{x_i} and γ_{y_j} disagree on all inputs.

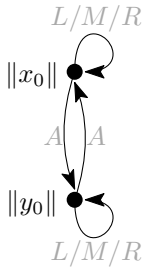


Figure 15: Contraction $\|\mathcal{X}\|$ of the CA $\mathcal{X}$ of Figure 14 and the proof of Lemma 5.27.

Thus the contraction of $\mathcal{X}$ (see Figure 15) has two classes, $\|x_0\|$ for the x_i's and $\|y_0\|$ for the y_i's, with the same local rules $\|\gamma\|_{\|x_0\|} = \mathsf{const}_\bullet$ and $\|\gamma\|_{\|y_0\|} = \mathsf{const}_\circ$ respectively. However, these two classes have the same neighbourhood quotients:

$$N_{\|x_0\|} = N_{\|y_0\|} = \{|\varepsilon, L, M, R|, |A|\},$$

and thus both the conditions of neighbourhood-uniformity and extension-uniformity would require $\|x_0\|$ and $\|y_0\|$ to have the same local rule if $\|X\|$ were to be a CA. □

5.3.2 Construction of the terminal coalgebra

The previous discussion showed that there is a unique homomorphism from every $\mathcal{X} \in \mathbf{CCA}_{\mathsf{G}}^{\Sigma}$ to $\|\mathcal{X}\|$. The strategy is now to construct the terminal coalgebra by putting all contractions of CA together in one coalgebra. To do so, two remaining challenges need to be solved:

1. There may (and in fact do) exist morphisms between contractions, and hence different morphisms from $\mathcal{X}$ to the union of all contractions. This is shown in Section 5.3.3.
2. The disjoined union of coalgebras $\biguplus_{\mathcal{X} \in \mathbf{CCA}_{\mathsf{G}}^{\Sigma}} \|\mathcal{X}\|$ quantifies over a collection greater than **Set**; the resulting collection is therefore not a set, and hence not the carrier of a coalgebra (since $C_{\mathsf{G}} \colon \mathbf{Set} \to \mathbf{Set}$). We will instead only work with contractions that are unique up to isomorphism, i.e., $\biguplus_{\mathcal{X} \in \mathbf{CCA}_{\mathsf{G}}^{\Sigma}/\cong} \|\mathcal{X}\|$, but it is still not obvious that this is a set. However, Section 5.3.4 will show that this union has a cardinality.

5.3.3 Morphisms between embeddings

To analyse challenge 1, we will show that the image of $\|(-)\|$ under $\mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ (denoted by $\|\mathbf{CCA}^{\Sigma}_{\mathsf{G}}\|$) is a thin category containing only embeddings as morphisms (i.e., there exists at most one embedding $\|\mathcal{X}\| \hookrightarrow \|\mathcal{Y}\|$), but also that not all morphisms are isomorphisms. Hence even after removing isomorphic contractions via a quotient, the union $\mathcal{T} = \biguplus_{\|\mathcal{X}\| \in \|\mathbf{CCA}^{\Sigma}_{\mathsf{G}}\|/\cong} \|\mathcal{X}\|$ of all (up to isomorphism unique) contractions still contains distinct BE cells; but – provided T it is a set – we can contract all the morphisms by using $\|\mathcal{T}\|$ to obtain a coalgebra towards which every CA has a unique morphism. The following four lemmas and theorem will justify those claims.

Lemma 5.28

For $\|\mathcal{X}\|, \|\mathcal{Y}\| \in \|\mathbf{CCA}^{\Sigma}_{\mathsf{G}}\|$, every $e\colon \|\mathcal{X}\| \to \|\mathcal{Y}\|$ is an embedding.

Proof. If h is not injective, then there would exist $\|x\| \neq \|x'\|$ such that $h(\|x\|) = h(\|x'\|)$. But then $\|x\| \overset{\mathsf{BE}}{\approx} \|x'\|$, contradicting Lemma 5.22. □

The next lemma is needed to show that there exists at most one embedding between contractions. The intuition of this next lemma is as follows: given two CA cells z_1 and z_2 that are essentially the same, and also $\mathscr{M} \circledS z_1$ is identical to $\mathscr{M} \circledS z_2$, then we can exchange all cells in $\mathscr{M} \circledS z_1$ with the corresponding cells $\mathscr{M} \circledS z_2$ without any essential change to the CA.

More generally, the next lemma will show that for a coalgebra $\mathcal{Y} = (Y, \langle \xi, \delta \rangle)$ containing two isomorphic subcoalgebras with isomorphism $\psi\colon \mathcal{Z}_1 \xrightarrow{\sim} \mathcal{Z}_2$, we can construct a coalgebra $\mathcal{Y}'$ with $\mathcal{Z}_1$ and $\mathcal{Z}_2$ exchanged, and "lift" ψ to:

$$\begin{aligned} &\psi^+\colon \mathcal{Y} \to \mathcal{Y}' \\ &\psi^+(y) \overset{\text{def}}{=} \begin{cases} \psi(z_1) \in Z_2 & u = z_1 \in Z_1 \\ \psi^{-1}(z_2) \in Z_1 & u = z_2 \in Z_2 \\ y & u = y \in Y \setminus (Z_1 \cup Z_2) \end{cases} . \end{aligned} \tag{44}$$

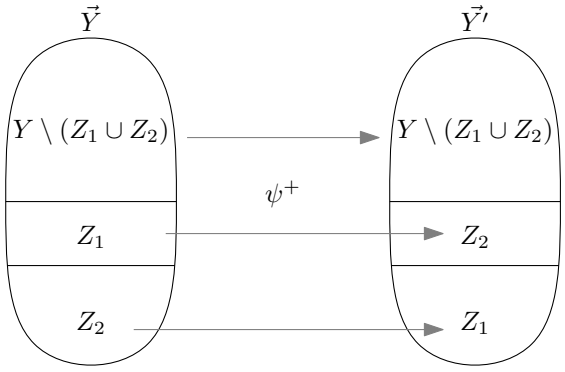


Figure 16: If $\mathcal{Y}$ has two isomorphic subcoalgebras $\mathcal{Z}_1$ and $\mathcal{Z}_2$, then there exists an isomorphism $\psi^+\colon \mathcal{Y} \cong \mathcal{Y}'$ that exchanges these subcoalgebras.

Lemma 5.29

The image $\mathcal{Y}' \overset{\text{def}}{=} \psi^+[\mathcal{Y}]$ is a coalgebra and $\psi^+ \colon \mathcal{Y} \xrightarrow{\sim} \mathcal{Y}'$ an isomorphism.

Proof. We first show that $\mathcal{Y}' = (Y', \langle \xi', \delta' \rangle)$ is a well-defined C_{G}-coalgebra.

Carrier: On the level of sets, it is clear that ψ^+ is a bijection $Y \xrightarrow{\sim} Y$, so the carrier Y' of $\mathcal{Y}'$ is simply Y.

ξ' of dynamics: As for the ξ' of $\mathcal{Y}'$, note that every element of Y' can be written in the form $\psi^+(u)$ for an $u \in Y$, so define for all $m \in \mathscr{M}$:

$$\begin{aligned} m \mathbin{⊚}' (-) &\colon Y' \to Y' \\ m \mathbin{⊚}' \psi^+(u) &\overset{\text{def}}{=} \psi^+(m \mathbin{⊚} u). \end{aligned} \tag{45}$$

It indeed satisfies the path-comonad laws (Eq. (11) and (12)):

$$\begin{aligned} \mathbf{1} \mathbin{⊚}' \psi^+(u) &\overset{\text{def}}{=} \psi^+(\mathbf{1} \mathbin{⊚} u) = \psi^+(u). \\ (m \bullet n) \mathbin{⊚}' \psi^+(u) &\overset{\text{def}}{=} \psi^+((m \bullet n) \mathbin{⊚} u) = \psi^+(m \mathbin{⊚} n \mathbin{⊚} u) \overset{\text{def}}{=} m \mathbin{⊚}' \psi^+(n \mathbin{⊚} u) \overset{\text{def}}{=} m \mathbin{⊚}' n \mathbin{⊚}' \psi^+(u). \end{aligned}$$

δ' of dynamics: Being an isomorphism, we have $N_{z_1} = N_{\psi(z_1)}$ for all $z_1 \in Z_1$, thus the extension property (Theorem 5.9) gives $\delta_{z_1} = \delta_{\psi(z_1)}$, and by definition of ψ^+, the latter indeed equals $\delta_{\psi^+(z_1)}$. Analogously $\delta_{z_2} = \delta_{\psi^{-1}(z_2)} = \delta_{\psi^+(z_2)}$ for all $z_2 \in Z_2$. Trivially also $\delta_y = \delta_{\psi^+(y)}$ for all $y \in Y \setminus (Z_1 \cup Z_2)$ (since $\psi^+(y) \overset{\text{def}}{=} y$), thus the δ' of $\mathcal{Y}'$ is just δ.

ψ^+ is a coalgebra isomorphism: That ψ^+ is a $\mathscr{M}$-ξ-ξ'-homomorphism follows directly from the definitions of δ' and ξ'. To show it is an isomorphism, we construct its inverse by lifting ψ^{-1} analogously:

$$\begin{aligned} (\psi^{-1})^+ &\colon \mathcal{Y}' \to \mathcal{Y} \\ (\psi^{-1})^+(u) &\overset{\text{def}}{=} \begin{cases} \psi^{-1}(z_1) \in Z_2 & u = z_1 \in Z_1 \\ \psi(z_2) \in Z_1 & u = z_2 \in Z_2 \\ y & u = y \in Y \setminus (Z_1 \cup Z_2) \end{cases}. \end{aligned}$$

Before we prove that $(\psi^{-1})^+$ has the properties of a coalgebra homomophism, we first show that it is indeed the inverse of ψ^+ on cells. For any $u \in Y$ we have:

$$\begin{aligned} (\psi^{-1})^+(\psi^+(u)) &\overset{\text{def}}{=} \begin{cases} u & u \in Y \setminus (Z_1 \cup Z_2) \\ \psi^{-1}(\psi(u)) & u \in Z_1 \\ \psi(\psi^{-1}(u)) & u \in Z_2 \end{cases} \\ &= u. \end{aligned}$$

Analogously we find $\psi^+((\psi^{-1})^+(u)) = u$ for all $u \in Y'$.

As usual, to prove that $(\psi^{-1})^+$ is a coalgebra homomorphism we will use Theorem 5.9. To see that $(\psi^{-1})^+$ is a $\mathscr{M}$-ξ'-ξ-homomorphism, again note that every element of Y' can be written in the form $\psi^+(u)$ for some $u \in Y$. Thus it suffices to take arbitrary $m \in \mathscr{M}$ and $u \in Y$ and observe that:

$$\begin{aligned} (\psi^{-1})^+(m \mathbin{⊚}' \psi^+(u)) &= (\psi^{-1})^+(\psi^+(m \mathbin{⊚} u)) \\ &= m \mathbin{⊚} u && \text{// } (\psi^{-1})^+ \text{ and } \psi^+ \text{ are inverses, so also on the cell } m \mathbin{⊚} u. \\ &= m \mathbin{⊚} (\psi^{-1})^+(\psi^+(u)). && \text{// Idem but for the cell } u. \end{aligned}$$

Writing $u' := \psi^+(u)$ this can be summarised as:

$$(\psi^{-1})^+(m \circledS {}'u') = m \circledS (\psi^{-1})^+(u'),$$

exactly the required relation.

It remains to show the extension property of $(\psi^{-1})^+$. But indeed, for an arbitrary $\psi^+(u) \in Y'$:

$$\delta_{(\psi^{-1})^+(\psi^+(u))} = \delta_u = \delta_{\psi^+(u)}$$

where the first equation follows from the fact that $(\psi^{-1})^+$ and ψ^+ are inverses on cells, and the second is the earlier observed extension property of ψ^+. □

Lemma 5.30

For $\|\mathcal{X}\|, \|\mathcal{Y}\| \in \|\mathbf{CCA}^{\Sigma}_{\mathsf{G}}\|$, there exist *at most one* embedding $e\colon \|\mathcal{X}\| \hookrightarrow \|\mathcal{Y}\|$.

Proof. Let $e_1, e_2\colon \|\mathcal{X}\| \hookrightarrow \|\mathcal{Y}\|$, then we need to show that $e_1 = e_2$.

Since both e_i are surjective on $\mathscr{M} \circledS e_i(\|x\|)$ for all $\|x\| \in \|X\|$ (by Lemma 5.4), they are isomorphisms $e_i\colon \|\mathcal{X}\| \xrightarrow{\sim} e_i[\|\mathcal{X}\|]$ (in $\mathsf{Coalg}(C_{\mathsf{G}})$). Surjectivity also means that no cell in $\|\mathcal{Y}\| \setminus e_i[\|\mathcal{X}\|]$ is reachable from a cell in $e_i[\|\mathcal{X}\|]$. Hence $e_i[\|\mathcal{X}\|]$ is a subcoalgebra of $\|\mathcal{Y}\|$ with only *incoming* paths from cells in $\|\mathcal{Y}\| \setminus e_i[\|\mathcal{X}\|]$.

In particular we have $e_1[\|\mathcal{X}\|] \cong \|X\| \cong e_2[\|X\|]$. Denote this isomorphism by $\psi\colon e_1[\|\mathcal{X}\|] \xrightarrow{\sim} e_2[\|\mathcal{X}\|]$. Using Lemma 5.29 we lift ψ to $\psi^+\colon \|\mathcal{Y}\| \xrightarrow{\sim} \|\mathcal{Y}\|'$, where $\|\mathcal{Y}\|'$ is $\|\mathcal{Y}\|$ but with subcoalgebras $e_1[\|\mathcal{X}\|]$ and $e_2[\|\mathcal{X}\|]$ exchanged. The construction of lemma 5.29 shows that $\psi^+ \restriction e_1[\|X\|]$ is just the isomorphism ψ with image $e_2[\|X\|]$ in $\|\mathcal{Y}\|'$, which allows us to define the composite isomorphism $\phi \stackrel{\text{def}}{=} e_2^{-1} \circ (\psi^+ \restriction e_1[\|X\|]) \circ e_1$.

$$\begin{array}{ccc} \|\mathcal{X}\| & \xRightarrow{\quad\phi\quad} & \|\mathcal{X}\| \\ {\scriptstyle e_1}\Downarrow & & \Uparrow{\scriptstyle e_2^{-1}} \\ e_1[\|\mathcal{X}\|] & \xRightarrow[\psi^+ \restriction e_1[\|X\|]]{} & e_2[\|\mathcal{X}\|] \end{array}$$

But $\|\mathcal{X}\|$ has no distinct BE cells (Lemma 5.22), thus all endohomomorphism must send every cell to itself. Hence $\phi = \mathsf{Id}_{\|\mathcal{X}\|}$.

Now take any $\|x\| \in \|\mathcal{X}\|$. Then

$$e_2^{-1}(\psi^+(e_1(\|x\|))) = \phi(\|x\|) = \|x\|.$$

Applying e_2 to both sides gives

$$\psi^+(e_1(\|x\|)) = e_2(\|x\|),$$

which is an equality of elements in the subcoalgebra $e_2[\|\mathcal{X}\|]$ of $\|\mathcal{Y}\|'$. Thus these elements are also equal in $\|\mathcal{Y}\|'$. But since ψ^+ is an isomorphism (this is the observation for which Lemma 5.29 is really used) ψ^+ has an inverse, and applying the inverse to both sides gives:

$$(\psi^+)^{-1}(\psi^+(e_1(\|x\|))) = (\psi^+)^{-1}(e_2(\|x\|)),$$

which is an equality in $\|\mathcal{Y}\|$. Choosing $h := (\psi^+)^{-1} \circ \psi^+$ and $k := (\psi^+)^{-1}$ shows that $h(e_1(\|x\|)) = k(e_2(\|x\|))$, thus $e_1(\|x\|) \overset{\mathsf{BE}}{\approx} e_2(\|x\|)$. But $\|\mathcal{Y}\|$ is a contraction, and therefore has no distinct BE cells. Thus $e_1(\|x\|) = e_2(\|x\|)$. Since $\|x\|$ was arbitrary, this proves $e_1 = e_2$. □

The next lemma will justify the need of contracting $\mathcal{T}$, since it may have non-trivial endohomomorphisms otherwise. It would be possible to postcompose such endohomomorphisms with arbitrary homomorphsim $h\colon \mathcal{X} \to \mathcal{T}$, which would show that h is not the unique homomorphism from $\mathcal{X}$ to $\mathcal{T}$.

Lemma 5.31

There exist $\|\mathcal{X}\|, \|\mathcal{Y}\| \in \|\mathbf{CCA}_{\mathsf{G}}^{\Sigma}\|$ and $e\colon \|\mathcal{X}\| \hookrightarrow \|\mathcal{Y}\|$ such that e is not an isomorphism.

Proof. Let the monoid $\mathscr{M}$ be freely generated over the alphabet $\{L, M, R, U\}$, and consider the contracted coalgebras $\|\mathcal{X}\|$ and $\|\mathcal{Y}\|$ as depicted in Figure 17. We can check that this are indeed contractions for a suitable choices of the local rules: all cells have different neighbourhood quotients that are incomparable under the $\mathsf{PartOrd}(N)$ ordering (Definition 2.2), and hence mutually incompatible (Definition 4.2).

$$\begin{aligned} N_{\|x_0\|} = N_{\|y_0\|} &= \{|L, U|, |\varepsilon, M, R|\} \\ N_{\|x_1\|} = N_{\|y_1\|} &= \{|\varepsilon, L, M|, |R, U|\} \\ N_{\|y_2\|} &= \{|\varepsilon|, |L|, |M, U|, |R|\} \\ N_{\|y_3\|} &= \{|\varepsilon, L, R|, |M, U|\} \end{aligned}$$

So the local rules at every cell can be chosen independently and differently. We choose them as such; then no pair of cells within $\|\mathcal{X}\|$ or withing $\|\mathcal{Y}\|$ is behavioural equivalent to another cell in the same coalgebra.

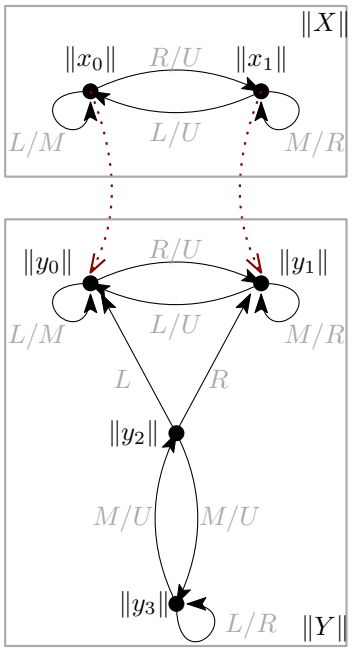


Figure 17: Embedding of a contracted coalgebra $\|\mathcal{X}\|$ (top) into a larger contracted coalgebra $\|\mathcal{Y}\|$ (bottom). The dotted arrows indicate the cell-map of the embedding. The underlying monoid $\mathscr{M}$ is freely generated over $\{L, M, R, U\}$. Only paths in the neighbourhood $N = \{\varepsilon, L, M, R, U\}$ are shown, but this is a generating set of $\mathscr{M}$. The notation M/U denotes that paths $M, U \in N$ lead to the same cell.

□

Remark 5.32. Any embedding $e\colon \|\mathcal{X}\| \hookrightarrow \|\mathcal{Y}\|$ has in its image all cells of $\mathscr{M} \copyright e(\|x\|)$ (for all $\|x\| \in \|X\|$), because of Lemma 5.4. Thus the embedding $e[\|\mathcal{X}\|]$ has exactly the same form as $\|\mathcal{X}\|$ and no outgoing arrows to other cells in $\|\mathcal{Y}\|$, but as the proof of Lemma 5.31 shows, it may have incoming arrows.

The next theorem will be used to prove the uniqueness of the homomorphism towards $\|\mathcal{T}\|$:

Theorem 5.33

There exists at most one unique coalgebra homomorphism $h\colon \mathcal{X} \to \|\mathcal{Y}\|$ when $\mathcal{X}, \mathcal{Y} \in \mathbf{CCA}^{\Sigma}_{\mathsf{G}}$.

Proof. It suffices to factor h via $\iota_{\mathcal{X}}\colon \mathcal{X} \to \|\mathcal{X}\|$ and an embedding $e\colon \|\mathcal{X}\| \hookrightarrow \|\mathcal{Y}\|$, since by Theorem 5.26 and Lemma 5.30 these two maps are necessary unique.

By naturality of ι w.r.t. the functor $\|(-)\|$ (Corollary 5.24) the following solid square commutes:

$$\begin{array}{ccc} \mathcal{X} & \xrightarrow{h} & \|\mathcal{Y}\| \\ {\scriptstyle \iota_{\mathcal{X}}}\downarrow & & \downarrow{\scriptstyle \iota_{\|\mathcal{Y}\|}} \quad \overset{\iota_{\|\mathcal{Y}\|}}{\uparrow} \\ \|\mathcal{X}\| & \xrightarrow{\|h\|} & \|\|\mathcal{Y}\|\| \end{array}$$

Lemma 5.25 tells us that $\iota_{\|\mathcal{Y}\|}$ is an isomorphism, thus it has an inverse (the dotted arrow in the above diagram). Hence

$$\iota^{-1}_{\|\mathcal{Y}\|} \circ \|h\| \circ \iota_{\mathcal{X}} = h,$$

and therefore we can take $e := \iota^{-1}_{\|\mathcal{Y}\|} \circ \|h\|$. □

5.3.4 Cardinality of union of contractions

The next lemmas will establish that $\|\mathbf{CCA}^{\Sigma}_{\mathsf{G}}\|/{\cong}$ is a set, and thus that $\biguplus_{\|\mathcal{X}\| \in \|\mathbf{CCA}^{\Sigma}_{\mathsf{G}}\|/\cong} \|\mathcal{X}\|$ is also a set.

The first lemma will be used to bound the number of behaviourally distinct (as in not-behaviourally-equivalent) cells in a contraction: contractions have no two cells with essentially the same successors.

Lemma 5.34

Let $\mathcal{X} = (X, \langle \chi, \gamma \rangle) \in \mathbf{CCA}^{\Sigma}_{\mathsf{G}}$. Let $x, x' \in X$ be such that for all $m \in \mathscr{M}$,

$$N_{m \circledast x} = N_{m \circledast x'}$$
$$\gamma_{m \circledast x} = \gamma_{m \circledast x'}$$

then $x \overset{\mathsf{BE}}{\approx} x'$.

Proof. We construct a coalgebra $\mathcal{X}'$ by using a construction similar to a contraction (c.f. Definition 5.13), but we only contract pairs of cells of the form $m \circledast x$ and $m \circledast x'$ into an equivalence class (for every $m \in \mathscr{M}$).

So let $\mathcal{X}' \overset{\text{def}}{=} (X/{\sim}, \langle \chi', \gamma' \rangle)$ where $\sim \subseteq X \times X$ is the reflexive and symmetric closure of the relation $m \circledast x \sim m \circledast x'$ (for all $m \in \mathscr{M}$). Denote equivalence classes of $\sim$ by $|y|$ (for all $y \in X$), and note that equivalence classes contain either 1 or 2 cells.

Define χ' as follows:

$$n \circledast' |y| \overset{\text{def}}{=} |n \circledast y|$$

for all $n \in \mathscr{M}$ and $y \in X$. To see that this definition is independent of the representative of $|y|$, there are only two cases to consider:

1. $|y| = \{y\}$, in which independence of the choice of representative is trivial.

2. $|y| = |m \mathbin{⦶} x|$ for some $m \in \mathscr{M}$. Then well-definedness follows from the following observation:

$$n \mathbin{⦶}' |m \mathbin{⦶} x| = |n \mathbin{⦶} m \mathbin{⦶} x| = |(n \bullet m) \mathbin{⦶} x| = |(n \bullet m) \mathbin{⦶} x'| = |n \mathbin{⦶} m \mathbin{⦶} x'| = n \mathbin{⦶}' |m \mathbin{⦶} x'| .$$

To see that χ' is a cowriter-comonad-coalgebra, observe that:

$$n \mathbin{⦶}' \ell \mathbin{⦶}' |y| \overset{\text{def}}{=} n \mathbin{⦶}' |\ell \mathbin{⦶} y| \overset{\text{def}}{=} |n \mathbin{⦶} \ell \mathbin{⦶} y| = |(n \bullet \ell) \mathbin{⦶} y| \overset{\text{def}}{=} (n \bullet \ell) \mathbin{⦶} |y| .$$

The definition of γ' follows the same approach as $\|\gamma\|$:

$$\gamma'_{|y|} \overset{\text{def}}{=} \gamma_y$$

where:

$$u \colon S^{N_{|y|}} \to S^{N_y}$$
$$u(k)(m) \overset{\text{def}}{=} k(|m|).$$

Note that u is only well-defined when $N_{|y|} \leq N_y$ (in $\mathsf{PartOrd}(N)$, Definition 2.2). That is, if $|m|_y = |n|_y$ implies $|m|_{|y|} = |n|_{|y|}$. To prove this, assume $|m|_y = |n|_y$, that is, $m \mathbin{⦶} y = n \mathbin{⦶} y$. Then applying $|(-)|$ to both sides gives $|m \mathbin{⦶} y| = |n \mathbin{⦶} y|$, which by definition of χ' is equivalent to $m \mathbin{⦶}' |y| = n \mathbin{⦶}' |y|$, which is equivalent to $|m|_{|y|} = |n|_{|y|}$.

Finally, we let $\phi \colon \mathcal{X} \to \mathcal{X}'$ be the quotient map $\phi \colon y \mapsto |y|$. That this is a well-defined coalgebra homomorphism follows directly from the definitions of χ' and γ' (and Theorem 5.9). We further observe that $\phi(x) = \phi(x')$, thus we conclude that $x \overset{\mathsf{BE}}{\approx} x'$. □

We proceed now by counting cardinalities. Fix an arbitrary monoid $\mathscr{M}$, neighbourhood $N \subseteq \mathscr{M}$ and state set S, and denote their cardinalities as:

$$\mu = \mathsf{card}(\mathscr{M})$$
$$\nu = \mathsf{card}(N)$$
$$\sigma = \mathsf{card}(S).$$

The following lemma gives a (very coarse) upper bound on the number of behaviourally distinct cells a CA may have.

Lemma 5.35

For a CA $\mathcal{X} = (X, \langle \chi, \gamma \rangle) \in \mathbf{CCA}^{\Sigma}_{\mathsf{G}}$, the number of distinct equivalence classes of X under the BE-relation is upper bounded by

$$\kappa = 2^{\mu} \mu(2^{2^{\nu}})(\sigma^{\sigma^{\nu}}).$$

Proof. From Lemma 5.34 it follows that a sufficient condition for cells $x, x' \in X$ to be BE is having:

- The same number of $\mathscr{M}$-reachable successors.
- The same $N_{m \mathbin{⦶} x} = N_{m \mathbin{⦶} x'}$ and $\gamma_{m \mathbin{⦶} x} = \gamma_{m \mathbin{⦶} x'}$ for all $m \in \mathscr{M}$.

The number of disjoined partitions of N is upper bounded by $2^{2^{\nu}}$, the number of subsets of subsets of N. Hence, there are at most $2^{2^{\nu}}$ choices for N_x for a given cell $x \in X$.

Recall that $\gamma_x \in S^{S^{N_x}}$, and since $\mathsf{card}(N_x) \leq \nu$, the number of choices of γ_x is bounded by $\sigma^{\sigma^{\nu}}$.

From the above two observations, it follows that there are at most $(2^{2^\nu})(\sigma^{\sigma^\nu})$ distinct choices of pairs (N_x, γ_x).

A cell x has at least 1 successor (itself) and at most μ successors. For a chosen number $1 \leq \alpha \leq \mu$ of successors of x, there are $\alpha(2^{2^\nu})(\sigma^{\sigma^\nu})$ ways to choose the data inside the successors. But by the cowriter-comonad structure (Eq. (12)), the successors of successors of x are also successors of x, hence this fixes the successors of all cells $m \mathbin{⊛} x$ automatically as well. Since there are 2^μ choices for α, and since $\alpha \leq \mu$, this means that the total number of ways to configure x and all reachable successors is bounded by

$$\kappa \stackrel{\text{def}}{=} 2^\mu \mu (2^{2^\nu})(\sigma^{\sigma^\nu}).$$

□

Note that a cell x may still have a myriad of (possibly independent) parents that it cannot see. The trick is that those parents themselves are also among those κ choices.

Corollary 5.36

If $\mathcal{X} \in \mathbf{CCA}^\Sigma_\mathsf{G}$ then $\|\mathcal{X}\|$ has at most κ cells.

Proof. $\|\mathcal{X}\|$ has no distinct behaviourally equivalent cells, and there are at most κ behaviourally distinct cells by Lemma 5.35. □

Corollary 5.37

There are, up to isomorphism, at most 2^κ distinct contractions of coalgebras in $\mathbf{CCA}^\Sigma_\mathsf{G}$.

Proof. This is the number of subsets of κ, the upper bound of the total number of possible cells. □

Let $A = \{\|\mathcal{X}\| \mid \mathcal{X} \in \mathbf{CCA}^\Sigma_\mathsf{G}\}/\cong$.

Theorem 5.38

$T \stackrel{\text{def}}{=} \biguplus_{\|\mathcal{X}\| \in A} \|\mathcal{X}\|$ is a set, and so is $\|T\|$. Therefore $\|\mathcal{T}\|$ is the terminal coalgebra in $\mathbf{CCA}^\Sigma_\mathsf{G}$.

Proof. The fact that T is a set follows from Corollary 5.37, which gives an upper bound on the cardinality of T.

Every $\mathcal{X} \in \mathbf{CCA}^\Sigma_\mathsf{G}$ has a unique morphism to $\|\mathcal{X}\|$ (Theorem 5.26), and clearly there is the inclusion map $i \colon \|\mathcal{X}\| \hookrightarrow T$. Thus we have the composition:

$$\mathcal{X} \xrightarrow{\iota_\mathcal{X}} \|\mathcal{X}\| \xrightarrow{\iota_{\|\mathcal{X}\|}} \||\mathcal{X}\|| \xrightarrow{\|i\|} \|T\|$$

Theorem 5.33 asserts that this coalgebra homomorphism is unique as well. □

5.4 Summary of coalgebra homomorphisms and behavioural equivalence

The concept of behavioural equivalence seems appropriate for describing the distinct atomic building blocks of a non-uniform CA (uniform CA have only one such building block), but not for reasoning about the global dynamics of a CA.

Using Theorem 5.38 as a practical characterisation of coalgebra homomorphisms, we have developed the concept of the *contraction* $\|\mathcal{X}\|$ of a CA $\mathcal{X}$. In a contraction, all BE cells are merged into one cell. There

is a unique map from a CA $\mathcal{X}$ to any contracted CA (Theorem 5.33), and also at most one map between contractions (Lemma 5.30). The reader acquainted with Homotopy Type Theory may see an analogy with propositional truncations: for every type A there is a unique map to its propositional truncation $\|A\|$, and there is at most one map between two propositional truncations (as they are propositions).

We used contractions to construct the terminal coalgebra $\|\mathcal{T}\|$ (Theorem 5.38). This coalgebra encodes a complete picture of all equivalence classes of BE cells of CA (defined for a given signature $(\mathscr{M}, N, S)$), as it contains exactly one cell for each class.

A modal logic obtained via the predicate-lifting framework for coalgebras cannot logically distinguish BE cells, and therefore it cannot distinguish a CA from its contraction. However, contractions preserve few of the global dynamics of a CA. The contraction of a uniform CA is particularly trivial, as it is a single cell (Lemma 5.11). For example, Example 5.12 shows that CGOL is contracted to the single-celled constant-dead-state CA; the logic would not be able to differentiate between the computationally universal dynamics of GCOL and a trivial constant state.

The conclusion is that a logic capable of describing spatio-temporal patterns in the trace sequence of a CA is necessarily also capable of distinguishing behaviourally equivalent cells. A logic generated by the predicate lifting framework is not able to do the latter. This justifies manually defining of a modal logic with semantics based on trace sequences, which will be the topic for the remainder of this thesis.

6 Logic for computation traces

Section 5 argued that the standard framework for defining modal logics for coalgebras due to Schröder [41] and Klin [22] is not appropriate for describing the global dynamics of a CA.

This section will manually construct an elementary modal logic for describing the states of cells in a trace of a CA. Like how Kripke modal logics are evaluated from the perspective of points, our logic will be evaluated locally from the perspective of cells. Given a coalgebra $\mathcal{X} = (X, \langle \chi, \gamma \rangle)$ on a monoid $\mathscr{M}$, we can assign a state to cells via a configuration $c_0 \colon X \to S$. Therefore it seems natural to take states as out analog of propositional letters and traces $[c_0, c_1, \dots]$ (which are entirely determined by c_0) as analogs to valuations. Hence our atomic formulas will be individual states $s \in S$ and models will be a coalgebras $\mathcal{X}$ paired with an initial configuration c_0.

In Kripke modal logics, when evaluating a formula at a point x, it is possible to specify that a subformula should be evaluated at a related point y via a diamond $\diamond_R$ if xRy. In the context of CA, there are two relation-like structures we can define diamonds for: the CA's connections between cells are encoded in χ, as well as the chain-shaped trace $c_0 \to c_1 \to c_2 \to c_3 \to \dots$. Therefore we will introduce "space" diamonds $\diamond_m$ for $m \in \mathscr{M}$ to traverse the CA's graph, as well as a "time" diamond $\boxtimes$ to evaluate a subformula at the next configuration in a trace. In combination with the familiar Boolean logic connectives $\wedge$, $\vee$, $\neg$ and $\to$, we can encode statements such as "*If my $m \in \mathscr{M}$ neighbour has state s_0, then I will have state s_1 in the next timestep*":

$$(\diamond_m s_0) \to \boxtimes s_1.$$

This section will incrementally build a basic modal language $\mathcal{L}_{\mathsf{stg}}$ that implements the ideas below. We will first construct the space diamonds, then add the time diamonds. Thereafter we discuss the possibilities of performing uniform substitution, and this section concludes with proving a collection of practical reasoning tools.

6.1 Space modalities

A configuration of a CA $\mathcal{X} = (X, \langle \chi, \gamma \rangle) \in \mathbf{CCA}_{\mathsf{G}}$ is an assignment $c \colon X \to S$ of each cell $x \in X$ to a state $s \in S$. This is essentially a Kripke model:

- The propositional letters are S, the worlds are X.
- There is a successor relation R_m for every $m \in \mathscr{M}$, namely xR_mx' iff $x' = m \circledast x$.
- The valuation $V\colon X \to \wp(S)$ is just the configuration: $V(x) \stackrel{\text{def}}{=} \{c(x)\}$. Note that that every world is assigned *exactly one* letter in a valuation c.

This perspective allows us to introduce a propositional "space modalities" modal logic $\mathcal{L}_{\mathsf{s}}$ with modalities $\diamond_m$ for every $m \in \mathscr{M}$, that can express "space" properties of a CA. This logic cannot yet express the dynamics of a CA (which follow from the local rules γ_x), but it is a foundation to which we will incrementally extend with more symbols in subsequent subsections.

6.1.1 Grammar

The grammar of $\mathcal{L}_{\mathsf{s}}$ is as follows:

$$\Psi ::= \quad \neg\psi \;\big|\; \psi_1 \vee \psi_2 \;\big|\; \bot \;\big|\; s \;\big|\; \diamond_m$$

where $m \in \mathscr{M}$ and $s \in S$. The notational abbreviations $\wedge$, $\to$, $\leftrightarrow$ and $\top$ are defined as usual. The states s act as logic constants; the intuition is that a formula "s" is satisfied at a cell x in configuration c iff x is assigned state s (i.e., iff $c(x) = s$).

6.1.2 Semantics

As in Kripke semantics, we distinguish between static *frames* and *models*, which are assignments of values to "variables". In the context of CA, the natural choice of these variables are the initial states of the cells.

Definition 6.1: Frame

A *frame* $\mathbb{X} = (\mathscr{M}, N, S, X, \langle\chi, \gamma\rangle)$ is a coalgebra $\mathcal{X} = (X, \langle\chi, \gamma\rangle) \in \mathbf{CCA}_{\mathsf{G}}$ together with the monoid $\mathscr{M}$, neighbourhood $N \subseteq \mathscr{M}$ and state set S on which $\mathcal{X}$ is defined.

Definition 6.2: Model

A *model* $(\mathbb{X}, c_0)$ is a frame $\mathbb{X} = (\mathscr{M}, N, S, \mathcal{X})$ together with an initial configuration $c_0\colon X \to S$.

Definition 6.3: Space modalities semantics

Define *modal satisfaction* of a formula $\psi \in \Psi$ in a model $(\mathbb{X}, c_0)$ at a cell $x \in X$, to be the proposition (which is strictly either true or false):

$$\mathbb{X}, c_0, x \vDash \psi,$$

whose truth is given recursively as follows:

$$\begin{array}{ll}
\mathbb{X}, c_0, x \vDash \bot & \text{is always false} \\
\mathbb{X}, c_0, x \vDash s & \text{iff } c_0(x) = s \\
\mathbb{X}, c_0, x \vDash \neg\psi & \text{iff } \mathbb{X}, c_0, x \nvDash \psi \\
\mathbb{X}, c_0, x \vDash \psi_1 \vee \psi_2 & \text{iff } (\mathbb{X}, c_0, x \vDash \psi_1 \text{ or } \mathbb{X}, c_0, x \vDash \psi_2) \\
\mathbb{X}, c_0, x \vDash \diamond_m\psi & \text{iff } \mathbb{X}, c_0, m \circledast x \vDash \psi
\end{array}$$

where $s \in S$, $m \in \mathscr{M}$, $\psi, \psi_1, \psi_2 \in \Psi$ and "$\mathbb{X}, c_0, x \nvDash \psi$" denotes "*it is false that* $\mathbb{X}, c_0, x \vDash \psi$".

A formula that holds in all choices of the initial configuration is called *valid*:

Definition 6.4: Validity of a formula

A formula ψ is *valid* in frame $\mathbb{X}$ at a cell $x \in X$, denoted $\mathbb{X}, x \vDash \psi$, if $\mathbb{X}, c_0, x \vDash \psi$ for all $c_0 \colon X \to S$. A formula ψ is *valid* in $\mathbb{X}$ itself, denoted $\mathbb{X} \vDash \psi$, if ψ is valid at all cells in X.

The *logic of a frame*, denoted $\mathsf{Log}(\mathbb{F})$, is the set of all formulas that a frame validates. The *logic of* x for a cell $x \in X$, denoted $\mathsf{Log}(x)$, is the set of formulas that are valid at x.

Stacking space diamonds, e.g., $\diamond_m \diamond_n$, semantically corresponds to monoid multiplication $n \bullet m$. An example usage of this property is that a logic for finitely generated monoids only need a diamond for every generating element. Formally, the property is:

Lemma 6.5

For all $\psi \in \Psi$, the formula $\diamond_m \diamond_n \psi$ is semantically equivalent to $\diamond_{n \bullet m} \psi$.

Proof. Directly from the definition of the semantics:

$$\begin{aligned} \mathbb{X}, c_0, x \vDash \diamond_m \diamond_n \psi \text{ iff } & \mathbb{X}, c_0, m \circledast x \vDash \diamond_n \psi \\ \text{iff } & \mathbb{X}, c_0, n \circledast m \circledast x \vDash \psi \\ \text{iff } & \mathbb{X}, c_0, (n \bullet m) \circledast x \vDash \psi \\ \text{iff } & \mathbb{X}, c_0, x \vDash \diamond_{n \bullet m} \psi \end{aligned}$$

□

6.1.3 Expressivity of space modalities

The logic $\mathcal{L}_{\mathsf{s}}$ can express both basic properties of the input configuration $c \colon X \to S$ and some properties of χ. As for a basic example:

Example 6.6. Consider the 2D grid (i.e., the monoid $\mathbb{Z}^2$), and denote the modalities corresponding to $(1,0), (-1,0), (0,1), (0,-1) \in \mathbb{Z}^2$ with $\diamond_{\rightarrow}, \diamond_{\leftarrow}, \diamond_{\uparrow}, \diamond_{\downarrow}$ respectively. As cells, simply use $X = \mathbb{Z}^2$ as well. Use the colours green, orange and purple as states (i.e., $S = \{s_{\mathsf{or}}, s_{\mathsf{gr}}, s_{\mathsf{pu}}\}$).

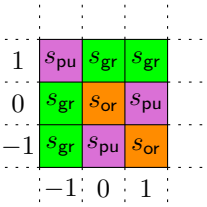


Figure 18: Part of a configuration $S \to \mathbb{Z}^2$ using the three-colour state set $S = \{s_{\mathsf{or}}, s_{\mathsf{gr}}, s_{\mathsf{pu}}\}$.

In the configuration c of Figure 18, and the frame $\mathbb{F} = (\mathbb{Z}^2, S, \overrightarrow{\mathbb{Z}^2})$, the point $x = (0,0)$ satisfies and unsatisfies

respectively the following formulas:

$$\mathbb{F}, c, x \vDash s_{\mathsf{or}}$$
$$\mathbb{F}, c, x \nvDash s_{\mathsf{gr}}$$
$$\mathbb{F}, c, x \vDash \Diamond\!\!\rightarrow\!\!\Diamond\, s_{\mathsf{or}}$$
$$\mathbb{F}, c, x \nvDash \Diamond\!\uparrow\, s_{\mathsf{pu}}$$
$$\mathbb{F}, x, x \vDash (\Diamond\!\!\leftarrow\, s_{\mathsf{gr}}) \wedge ((\Diamond\!\uparrow\, s_{\mathsf{pu}}) \vee (\neg\, \Diamond\!\downarrow\, s_{\mathsf{s}_{gr}})).$$

By using the notion of *validity*, the logic $\mathcal{L}_{\mathsf{s}}$ can already separate cells that are BE, implying $\mathcal{L}_{\mathsf{s}}$ is more expressive than the notion of BE. A frame $\mathbb{F}$ *validates* $\psi \in \Psi$, denoted as $\mathbb{F} \vDash \psi$, if $\mathbb{F}, c, x \vDash \psi$ for all $c\colon X \to S$ and $x \in X$. The *logic of a frame*, denoted $\mathsf{Log}(\mathbb{F})$, is the set of formulas that the frame validates. Analogously a cell $x \in X$ may validate a ψ if $\mathbb{F}, c, x \vDash \psi$ for all $c\colon X \to S$, which in turn defines the logic of a cell $\mathsf{Log}(x)$.

Lemma 6.7

There exist $\mathcal{X}, \mathcal{Y} \in \mathbf{CCA}_{\mathsf{G}}$ and $x \in X$ and $y \in Y$ such that $x \overset{\mathsf{BE}}{\approx} y$ but $\mathsf{Log}(x) \neq \mathsf{Log}(y)$.

Proof. Refer back to Example 5.12, which showed that all cells in CGOL are BE to the CA with a single cell and the local rule that always outputs the "dead" state $\circ$. Denote this single-cell-constant-dead CA by $\mathbb{1}_\circ$, with the single cell $*$, state set $\{\bullet, \circ\}$ and $\gamma_* = \mathsf{const}_\circ$. The CA $\mathbb{1}_\circ$ uses the same signature $(\mathbb{Z}^2, N, S = \{\bullet, \circ\})$ as $\overrightarrow{\mathsf{CGOL}}$ (coalgebra homomorphisms only exist between CA defined on the same signature) where $N = \{(i,j) \mid i,j \in \{-1, 01\}\}$. Now a configuration on $\mathbb{1}_\circ$ is of type $c\colon \{*\} \to \{\bullet, \circ\}$ and all neighbours of $*$ (which is only $*$ itself!) all receive the same state in any configuration. Hence

$$\circ \to \Diamond\!\uparrow\, \circ \in \mathsf{Log}(*).$$

In CGOL this formula is not *valid* at any cell $(x, y) \in \mathbb{Z}^2$, seen (for example) by the configuration $c(x, y) = \circ$ and $c(x, y+1) = \bullet$. (Note that the formula is *satisfied* in the *model* of CGOL's frame with a configuration c s.t. $c(x, y) = c(x, y+1)$. But this does not hold for all configurations on $\overrightarrow{\mathsf{CGOL}}$). □

Example 6.8 (Neighbourhood quotients)**.** The logic $\mathcal{L}_{\mathsf{s}}$ can express properties of the neighbourhood quotients. For example the formula

$$\Diamond_n s_0 \leftrightarrow \Diamond_m s_0 \tag{46}$$

is valid at a cell x iff $|n|_x = |m|_x$ (assuming $S \ncong \mathbb{1}$). This can be seen as follows: if $n \circledS x \neq m \circledS x$, then one can define a configuration such that $c(n \circledS x) \neq c(m \circledS x)$, in which case (46) is not satisfied.

6.2 Time modalities

The logic $\mathcal{L}_{\mathsf{s}}$ cannot express any properties about the dynamics of a CA; neither local dynamic properties such as the local rules γ_x (the semantics c.f. Definition 6.3 are independent of γ_x) nor properties about the global dynamics of a CA. Recall from the definition of a general CA (Definition 4.6) that the global dynamics are a sequence of configurations $(c_i)_{i \in \mathbb{N}}$ (where $c_i\colon X \to S$) computed by recursively applying the global rule $G_{\mathbb{X}}$ (Eq. (28)) to an configuration c_0.

In order to express the evolution of the configurations over time we introduce a new *time modality* $\boxtimes$ to the logic $\mathcal{L}_{\mathsf{s}}$, obtaining the "space & time modalities" logic $\mathcal{L}_{\mathsf{st}}$, with semantics:

Definition 6.9: Space & time modalities semantics

Define *modal satisfaction* of a formula $\psi \in \Psi$ in a model $(\mathbb{X}, c_0)$ at a cell $x \in X$, to be the proposition (which is strictly either true or false):

$$\mathbb{X}, c_0, x \vDash \psi.$$

Unlike in the semantics of $\mathcal{L}_{\mathsf{s}}$, modal satisfaction depends depends recursively on the truth in models $(\mathbb{X}, G^t_{\mathbb{X}}(c_0))$. Thus we define the truth of propositions

$$\mathbb{X}, G^t_{\mathbb{X}}(c_0), x \vDash \psi \tag{47}$$

for all $t \in \mathbb{N}$. Since c_0 is usually clear from context, we ease the notation by writing c_t for $G^t_{\mathbb{X}}(c_0)$. Now the truth of (47) can be given recursively as:

$$\begin{aligned}
&\mathbb{X}, c_t, x \vDash \bot && \text{is always false}\\
&\mathbb{X}, c_t, x \vDash s && \text{iff } c_t(x) = s\\
&\mathbb{X}, c_t, x \vDash \neg\psi && \text{iff } \mathbb{X}, c_t, x \nvDash \psi\\
&\mathbb{X}, c_t, x \vDash \psi_1 \vee \psi_2 && \text{iff } (\mathbb{X}, c_t, x \vDash \psi_1 \text{ or } \mathbb{X}, c_t, x \vDash \psi_2)\\
&\mathbb{X}, c_t, x \vDash \diamond_m \psi && \text{iff } \mathbb{X}, c_t, m \circledcirc x \vDash \psi\\
&\mathbb{X}, c_t, x \vDash \boxtimes\psi && \text{iff } \mathbb{X}, c_{t+1}, m \circledcirc x \vDash \psi
\end{aligned}$$

The only difference between the semantics of $\mathcal{L}_{\mathsf{st}}$ and $\mathcal{L}_{\mathsf{s}}$ (Definition 6.3) are the $\boxtimes$ case, and generalising from c_0 to all c_t.

6.2.1 Expressivity of time modalities

The logic $\mathcal{L}_{\mathsf{st}}$ gives a positive answer to our research question, but only for finite patterns:

Example 6.10. The logic $\mathcal{L}_{\mathsf{st}}$ can already express any finite spatio-temporal pattern of a trace of a rooted (Definition 3.8) CA. To see this, let $\mathbb{X} = (\mathscr{M}, N, S, X, \langle \chi, \gamma \rangle) \in \mathbf{CCA}_{\mathsf{G}}$ be a CA rooted in $r \in X$. For any cell $x \in X$ and state $s \in S$, we can express that x has state s in the $(t-1)^{\text{th}}$ configuration of a trace with initial configuration $c_0 \colon X \to S$ as follows. First we rewrite $x = m \circledcirc r$ (for some $m \in \mathscr{M}$, which is possible since r is the root), and then we note that $G^t_{\mathbb{X}}(x) = s$ holds iff $\mathbb{X}, c_0, r \vDash \boxtimes^t \diamond_m s$. For patterns involving multiple cells and/or multiple points in time we simply use conjunctions.

We will now show examples of properties related to the dynamics of a CA that $\mathcal{L}_{\mathsf{st}}$ can express but $\mathcal{L}_{\mathsf{s}}$ cannot.

Example 6.11. $\mathcal{L}_{\mathsf{st}}$ can logically distinguish a CA whose local rules are the constant-$\circ$-output from a CA whose local rules are the constant-$\bullet$-output; the latter validates $\boxtimes\bullet$ but not $\boxtimes\circ$, for the former it is exactly the other way around.

Lemma 6.12

If N is finite, then $\mathcal{L}_{\mathsf{st}}$ can express the mapping of γ_x piecewise for every input $z \colon N_x \to S$.
If S is also finite, then $\mathcal{L}_{\mathsf{st}}$ can express γ_x entirely in one formula.

Proof. The formula for $\gamma_x(z)$ is:

$$\left(\bigwedge_{n \in N} \diamond_n z|n|_x \right) \to \boxtimes\gamma_x(z). \tag{48}$$

The reading is as follows: $\bigwedge_{n\in N} \diamond_n z(|n|_x)$ is satisfied in $\mathbb{X}, c_t, x$ iff $c_t \restriction (N \circledS x) = z \circ |(-)|_x$, so it encodes *if the neighbours of x have the states as encoded in z.* The conclusion $\boxtimes\gamma_x(z)$ encodes that $\mathbb{X}, c_{t+1}, x \vDash \gamma_x(z)$, i.e., that x will have state $\gamma_x(z)$ in the next timestep.

If S is finite, then S^{N_x} is also finite, so one can take the conjunction over all $z \in S^{N_x}$ of equations (48). □

Example 6.13 (Local rule)**.** Suppose we have a CA $\mathcal{X}$ on the monoid freely generated over $\{L, R\}$ with the finite neighbourhood $N = \{\mathbf{1}, L, R\}$ and state set S. Given a cell x with a full neighbourhood $N_x = \{N\}$, we can express γ_x on an input $z\colon N \to S$ as:

$$\bigwedge_{z\colon N\to S} (z(\mathbf{1}) \wedge \diamond_L z(L) \wedge \diamond_R z(R)) \to \boxtimes\gamma_x(z \backslash\!\backslash x).$$

6.3 Global variables and uniform substitution

In the context of Kripke frames, a key strength of normal modal logics is the ability to characterise a class $\mathcal{C}$ of frames by a finite set of formules $\Psi_\mathcal{C}$ such that $\mathbb{X} \in \mathcal{C}$ iff $\mathbb{X} \vDash \psi$ for all $\psi \in \Psi_\mathcal{C}$. For example, $p \to \diamond p$ characterises frames based on a reflexive relation. This is possible because of the propositional letters $p \in Phi$ in Kripke semantics: these act as global variables and can be *uniformly substituted* by arbitrary subformulas without changing the validity of the outer formula.

This raises the question whether one can also introduce a form of uniform substitution in the logic for CA, as this would be helpful to finitely classify classes of CA.

The answer is positive, but we cannot simply use the states $s \in S$ as propositional letters. To see this, observe that Kripke propositional letters are indistinguishable and whose semantics are entirely determined by the valuation $V\colon W \to \wp(\Phi)$ added to a frame $\mathbb{X} = (W, R \subseteq W \times W)$ to construct a model $(\mathbb{X}, V)$. However, states $s \in S$ are not necessarily indistinguishable and may partially have a constant value that the choice of c_0 (our analog of V) cannot change. The simplest example is a constant local rule $\gamma_x = \mathsf{const}_{s_0}$ for some $s_0 \in S$. A model may freely choose c_0, but $c_t + 1(x) = s_0$ for all $t \in \mathbb{N}$. Hence x validates $\boxtimes s_0$. Obviously we cannot substitute s_0 by any arbitrary formula, such as "s_1" for a state $s_1 \neq s_0 \in S$!

The solution is straightforward: only consider the states in c_0, which do not have any constant value. We introduce new 0-ary symbols $\mathsf{v_S}$ for all $s \in S$ and extend the semantics with the rule:

$$\mathbb{X}, G^t_\mathbb{X}(c_0), x \vDash \mathsf{v_S} \text{ iff } c_0(x) = s. \tag{49}$$

When c_0 is clear from context, this will be written as $\mathbb{X}, c_t, x \vDash \mathsf{v_S}$ iff $c_0(x) = s$. We denote extension of $\mathcal{L}_{\mathsf{st}}$ with global variables as $\mathcal{L}_{\mathsf{stg}}$.

One complication remains: a normal modal logic formula ψ in a Kripke frame becomes either true or false after giving a valuation V. A formula can be seen as a function $\wp(\Phi)^W \to \mathbb{2}$ that maps a valuation to a true/false truth value. But for CA, the truth value of a formula ψ might also depend on the point in time in which it is evaluated. It follows that one cannot substitute a variable $\mathsf{v_S}$ (which does not depend on the point in time in which it is evaluated, i.e., $\mathbb{X}, c_0, x \vDash \mathsf{v_S}$ iff $\mathbb{X}, c_t, \vDash \mathsf{v_S}$ for all $t \in \mathbb{N}$) by a formula ψ whose truth does change over time: that is, if $\mathbb{X}, c_{t_1}, x \vDash \psi$ but $\mathbb{X}, c_{t_2}, x \nvDash \psi$ for some $t_1, t_2 \in \mathbb{N}$. However, when restricting to time invariant formulas, uniform substitution *does* remain sound for CA.

Definition 6.14: Time invariant formula

A formula $\psi \in \Psi$ is *time invariant* if, for all $c_0\colon X \to S$:

$$\mathbb{X}, x \vDash \psi \text{ iff for all } t \in \mathbb{N}\colon \mathbb{X}, x \vDash \boxtimes^t \psi.$$

Denote the time invariant formulas in Ψ of a cell by $\mathsf{TI}_\Psi(x)$, and the formulas that are time invariant in all cells in a CA $\mathcal{X}$ by $\mathsf{TI}_\Psi(\mathcal{X})$.

For a frame $\mathbb{X}$ based on $\mathcal{X}$ we will also write $\mathsf{TI}_\Psi(\mathbb{X})$ to denote the same set as $\mathsf{TI}_\Psi(\mathcal{X})$.

Lemma 6.15: Valid formulas are time invariant

If $\mathbb{X}, x \vDash \psi$ then ψ is time-invariant.

Proof. Take any $c_0 \colon X \to S$ and any $t \in \mathbb{N}$. We need to show that $\mathbb{X}, c_0, x \vDash \mathrm{x}^t\psi$, but this holds iff $\mathbb{X}, c_t, x \vDash \psi$, which indeed follows from the assumption $\mathbb{X}, x \vDash \psi$ (since $c_t = G^t_{\mathbb{X}}(c_0)$ is just another choice of model). □

Note that we cannot restrict the grammar to only time invariant formulas, since time invariance depends on the CA. For example, recall CGOL and $\mathbb{1}_\circ$ from the proof of Lemma 6.7 and Example 5.12. The formula $\circ \to \mathrm{x}\circ$ is not valid in CGOL but it is if valid (and hence time invariant) in $\mathbb{1}_\circ$.

Further note that the absence of constants $s \in S$ in a formula ψ is not required for time invariance. For example, $s \vee \top$ is always valid and $s \wedge \bot$ is always invalid, and hence both are time invariant.

Lemma 6.16: Uniform substitution of time invariant formulas

If $\mathbb{X} \vDash \psi$ (and $S \ncong \mathbb{1}$), then for all $s \in S$ and $\phi \in \mathsf{TI}_\Psi(\mathbb{X})$ it holds also that $\mathbb{X} \vDash \psi\left[{}^{\phi}/_{\mathsf{v}_s}\right]$.

It is most convenient to first prove the following auxiliary lemma:

Lemma 6.17

For all $\theta \in \Psi$ containing the term v_s, $s \neq s' \in S$, $\phi \in \mathsf{TI}_\Psi(\mathbb{X})$ and $c_0 \colon X \to S$, if we define

$$c_0' \overset{\text{def}}{=} \begin{cases} s & \mathbb{X}, c_0, x \vDash \phi \\ s' & \text{otherwise} \end{cases},$$

then for all $t \in \mathbb{N}$ and $x \in X$:

$$\mathbb{X}, c_t', x \vDash \theta \text{ iff } \mathbb{X}, c_t, x \vDash \theta\left[{}^{\phi}/_{\mathsf{v}_s}\right].$$

where we abbreviated $G^t_{\mathbb{X}}(c_0')$ as c_t'.

Proof. Proceed by structural induction on θ. We assumed that θ contains v_s, so the cases $\theta = s \in S$ and $\theta = \bot$ do not apply.

Case $\theta = \mathsf{v}_s$: By the semantics of v_s, $\mathbb{X}, c_t', x \vDash \mathsf{v}_s$ iff $c_0'(x) = s$. By definition of c_0', this holds iff $\mathbb{X}, c_0, x \vDash \phi$. The latter holds iff $\mathbb{X}, c_t, x \vDash \mathsf{v}_s\left[{}^{\phi}/_{\mathsf{v}_s}\right]$, as required.

Note that this first case is the reason for introducing v_s. If we would use the atoms $s \in S$ instead as propositional letters to be substituted, then we would only be able to show the equality for $t = 0$ in the $\theta = s$ case. However, we need it to show for all $t \in \mathbb{N}$ in order to have a sufficiently strong induction hypothesis for the $\theta = \mathrm{x}\psi$ case.

Case $\theta = \psi_1 \vee \psi_2$:
Induction hypothesis: for all $x \in X$, $t \in \mathbb{N}$ and $j \in \{1, 2\}$ it holds that:

$$\mathbb{X}, c_t', x \vDash \psi_j \text{ iff } \mathbb{X}, c_t, x \vDash \phi_j\left[{}^{\phi}/_{\mathsf{v}_s}\right].$$

Observe that $(\psi_1 \vee \psi_2)\left[{}^{a}/_{b}\right] = (\psi_1\left[{}^{a}/_{b}\right]) \vee (\psi_2\left[{}^{a}/_{b}\right])$ for all strings a and b. Therefore we can compute:

$$\begin{aligned}
&\mathbb{X}, c'_t, x \vDash \psi_1 \vee \psi_2 \\
&\qquad \text{iff } (\mathbb{X}, c'_t, x \vDash \psi_1 \text{ or } \mathbb{X}, c'_t, x \vDash \psi_2) \\
&\qquad \text{iff } (\mathbb{X}, c_t, x \vDash \psi_1\left[{}^{\phi}/_{\mathsf{vs}}\right] \text{ or } \mathbb{X}, c_t, x \vDash \psi_2\left[{}^{\phi}/_{\mathsf{vs}}\right]) && \text{// By the induction hypothesis.} \\
&\qquad \text{iff } \mathbb{X}, c_t, x \vDash (\psi_1 \vee \psi_2)\left[{}^{\phi}/_{\mathsf{vs}}\right].
\end{aligned}$$

Case $\theta = \neg\psi$:
Induction hypothesis: for all $x \in X$ and $t \in \mathbb{N}$ it holds that:

$$\mathbb{X}, c'_t, x \vDash \psi \text{ iff } \mathbb{X}, c_t, x \vDash \psi\left[{}^{\phi}/_{\mathsf{vs}}\right].$$

Now we compute:

$$\begin{aligned}
\mathbb{X}, c'_t, x \vDash \neg\psi &\text{ iff } \mathbb{X}, c'_t, x \nvDash \psi \\
&\text{iff } \mathbb{X}, c_t, x \nvDash \psi\left[{}^{\phi}/_{\mathsf{vs}}\right] \\
&\text{// By the induction hypothesis and the fact that a formula is either satisfied or not.} \\
&\text{iff } \mathbb{X}, c_t, x \vDash \neg(\psi\left[{}^{\phi}/_{\mathsf{vs}}\right]) \\
&\text{iff } \mathbb{X}, c_t, x \vDash (\neg\psi)\left[{}^{\phi}/_{\mathsf{vs}}\right].
\end{aligned}$$

Case $\theta = \boxtimes\psi$:
The induction hypothesis is as in the $\neg$ case.

$$\begin{aligned}
\mathbb{X}, c'_t, x \vDash \boxtimes\psi &\text{ iff } \mathbb{X}, c'_{t+1}, x \vDash \psi \\
&\text{iff } \mathbb{X}, c_{t+1}, x \vDash \psi\left[{}^{\phi}/_{\mathsf{vs}}\right] && \text{// By the induction hypothesis.} \\
&\text{iff } \mathbb{X}, c_t, x \vDash \boxtimes(\psi\left[{}^{\phi}/_{\mathsf{vs}}\right]) \\
&\text{iff } \mathbb{X}, c_t, x \vDash (\boxtimes\psi)\left[{}^{\phi}/_{\mathsf{vs}}\right].
\end{aligned}$$

Case $\theta = \diamond_m\psi$ $(m \in \mathscr{M})$:
The induction hypothesis is again as in the $\neg$ case.

$$\begin{aligned}
\mathbb{X}, c'_t, x \vDash \diamond_m\psi &\text{ iff } \mathbb{X}, c'_t, m \circledast x \vDash \psi \\
&\text{iff } \mathbb{X}, c_t, m \circledast x \vDash \psi\left[{}^{\phi}/_{\mathsf{vs}}\right] && \text{// By the induction hypothesis.} \\
&\text{iff } \mathbb{X}, c_t, x \vDash \diamond_m(\psi\left[{}^{\phi}/_{\mathsf{vs}}\right]) \\
&\text{iff } \mathbb{X}, c_t, x \vDash (\diamond_m\psi)\left[{}^{\phi}/_{\mathsf{vs}}\right].
\end{aligned}$$

□

Proof of Lemma 6.16. Assume that $\mathbb{X} \vDash \psi$ and that $\phi \in \mathsf{TI}_\Psi(\mathcal{X})$. It is to show that $\mathbb{X} \vDash \psi\left[{}^{\phi}/_{\mathsf{vs}}\right]$. To this end, take any arbitrary $c_0 \colon X \to S$ and $x \in X$, then it suffices to show that $\mathbb{X}, c_0, x \vDash \psi\left[{}^{\phi}/_{\mathsf{vs}}\right]$, i.e., that

$$\mathbb{X}, c_0, x \vDash \psi\left[{}^{\phi}/_{\mathsf{vs}}\right] \tag{50}$$

But define c'_0 as in Lemma 6.17, then $\mathbb{X} \vDash \psi$ implies that $\mathbb{X}, c'_0, x \vDash \psi$. Now Eq. (50) follows directly from Lemma 6.17. □

We conclude with an example application of global variables and time-invariant uniform substitution.

Example 6.18. Given a CA $\mathbb{X} = (\mathscr{M}, N, S, \mathcal{X})$ with at least two distinct states $s_0, s_1 \in S$, the formula

$$\mathsf{v}_{s_{s_0}} \to \diamond_m \mathsf{v}_{s_{s_0}} \tag{51}$$

is valid at a cell $x \in X$ iff $x = m \circledast x$ (otherwise we could unsatisfy with a configuration c_0 s.t. $c(x) = s_0$ and $c(m \circledast x) = s_1$). Uform substitution of time-invariant formulas allows to substitute $\mathsf{v}_{s_{s_0}}$ by any other time invariant formula. This is not surprising: if x satisfies ψ and $x = m \circledast x$, then $m \circledast x$ must clearly also satisfy ψ!

6.4 Logical reasoning rules

One of the main applications of defining a logic is the possibility to perform symbolic reasoning. Logical reasoning can, for example, be used to prove properties about the mathematical structures that form the semantics. This requires reasoning rules that are *sound*, i.e., that are consistent the semantics. We will present several such rules in the style of natural deduction, many of which will be used in proofs in later sections.

In this section, we use $\mathbb{X} = (\mathscr{M}, N, S, X, \langle \chi, \gamma \rangle)$ as an arbitrary frame. A rule about validity of a formula in $\mathbb{X}$ will be presented as follows:

Definition 6.19: Sound rule

A rule with n-premises $\mathbb{X} \vDash \phi_1, \mathbb{X} \vDash \phi_2, \ldots, \mathbb{X} \vDash \phi_n$ and conclusion $\mathbb{X} \vDash \psi$,
denoted $\dfrac{\mathbb{X} \vDash \phi_1 \quad \mathbb{X} \vDash \phi_2 \quad \ldots \quad \mathbb{X} \vDash \phi_n}{\mathbb{X} \vDash \psi}$, is *sound* if $\mathbb{X} \vDash \psi$ whenever $\mathbb{X} \vDash \phi_i$ for all $1 \leq i \leq n$.

Double bars, as in $\dfrac{A}{\;\;B\;\;}$, denote the pair of rules $\dfrac{A}{B}$ and $\dfrac{B}{A}$.

A rule may apply both to space diamonds $\diamond_m$ (for all choices of $m \in \mathscr{M}$) as well as for time diamonds $\boxtimes$, in which case the notation $\diamond$ will be used. All instances of $\diamond$ within a rule must be substituted with the same diamond.

We will also write metamathematical statements (such as $s \in S$) as premises in rules. Technically this means that we define a family of rules over all choices of the metavariables in the metamathematical premises.

Be aware that we give no guarantee of completeness: there may exist valid formulas that cannot be derived from the rules in this section.

The first two rules are very versatile; they show that we can perform basic propositional logic in $\mathcal{L}_{\mathsf{stg}}$.

Lemma 6.20

For any propositional logical tautology τ in propositional variables $x_1, x_2, \ldots x_n$, we have the rule:

$$\frac{\psi_1, \psi_2, \ldots, \psi_n \in \Psi}{\mathbb{X} \vDash \tau \left[{}^{\psi_1}/_{x_1}\right] \left[{}^{\psi_2}/_{x_2}\right] \ldots \left[{}^{\psi_n}/_{x_n}\right]}\ \mathsf{Taut}$$

Proof. The semantics for $\mathcal{L}_{\mathsf{stg}}$ concerning the symbols $\neg$, $\vee$ and $\bot$ (and hence also $\to$, $\wedge$ and $\top$) are equivalent to those of propositional logic.

In any model given by a configuration $c_0 \colon X \to S$, the semantics of the formulas ψ_i evaluate to `tt` or to `ff`. So the semantics in c_0 of a formula ϕ containing ψ_i are (in c_0) the same as $\phi \left[{}^{\phi_i}/_{\top}\right]$ or $\phi \left[{}^{\phi_i}/_{\bot}\right]$ respectively. Since τ is a tautology, it is true for any substitution of the variables $x_1, x_2, \ldots, x_n$ by $\top$ and $\bot$. □

Lemma 6.21: Modus Ponens

The following rule is sound:

$$\frac{\mathbb{X} \vDash \psi \to \phi \quad \mathbb{X} \vDash \psi \quad \psi, \phi \in \Psi}{\mathbb{X} \vDash \phi}\ \mathsf{MP}$$

Proof. Take any configuration $c_0 \colon X \to S$. Then for all $x \in X$ we have (unfolding the definition of $\to$) $\mathbb{X}, c_0, x \vDash \neg\psi \vee \phi$ and $\mathbb{X}, c_0, x \vDash \psi$. This combination is only possible if $\mathbb{X}, c_0, x \vDash \phi$. Since c_0 and x were arbitrary we find $\mathbb{X} \vDash \phi$. □

Remark 6.22. Many common reasoning strategies can be obtained as a combination of a propositional tautology and MP. For example, $\wedge$-elimination can be encoded as a tautology $A \wedge B \to A$. Thus, if one has derived that $\mathbb{X} \vDash \phi_A \wedge \phi_B$ for some $\phi_A, \phi_B \in \Psi$, we first obtain the tautology $\mathbb{X} \vDash \phi_A \wedge \phi_B \to \phi_A$ via Taut (Lemma 6.20), and conclude $\mathbb{X} \vDash \phi_A$ via MP (Lemma 6.21).

MP gives a tool for implication elimination, but implication introduction is also often useful. The following lemma shows that we can introduce implications by the usual method of propositional logic.

Lemma 6.23: Implication introduction

If one can prove $\mathbb{X}, c_0, x \vDash \phi$ when assuming $\mathbb{X}, c_0, x \vDash \psi$, then $\mathbb{X}, c_0, x \vDash \psi \to \phi$ holds in general.

Proof. Assume we can derive $\mathbb{X}, c_0, x \vDash \phi$ from $\mathbb{X}, c_0, x \vDash \psi$.

By definition of "$\to$" we know $\mathbb{X}, c_0, x \vDash \psi \to \phi$ holds iff $\mathbb{X}, c_0, x \vDash \neg\psi \vee \phi$. Assume $\mathbb{X}, c_0, x \nvDash \neg\psi$, then we need to show that $\mathbb{X}, c_0, x \vDash \phi$. But $\mathbb{X}, c_0, x \vDash \psi$ holds iff $\mathbb{X}, c_0, x \vDash \psi$ (by Lemma 6.29). By the premise of the lemma, this implies $\mathbb{X}, c_0, x \vDash \phi$. The latter was to be shown. □

The next two rules follow from the fact that, in any configuration, a cell has exactly one state.

Lemma 6.24: Single state

The following rule is sound:

$$\frac{s_0, s_1 \in S \quad s_0 \neq s_1}{\mathbb{X} \vDash s_0 \to \neg s_1}\ \mathsf{Det1}$$

Proof. Take any $x \in X$ and $c_0 \colon X \to S$. Suppose $s_0 \neq s_1 \in S$ and that $\mathbb{X}, c_0, c \vDash s_0$. Then, by the semantics of the formula s_0, it must be that $c_0(x) = s_0$. Thus $c_0(x) \neq s_1$, which implies $\mathbb{X}, c_0, x \nvDash s_1$. Hence $\mathbb{X}, c_0, x \vDash s_0 \to \neg s_1$. □

The following rule shows that we cannot distinguish $\diamond$ from $\neg\diamond\neg$; this is unlike in Kripke modal logics, where $\Box \stackrel{\text{def}}{=} \neg\diamond\neg$ does have different semantics than $\diamond$.

Lemma 6.25: Diamonds are boxes

For any $\psi \in \Psi$ the following rule is sound:

$$\frac{\mathbb{X} \vDash \diamond\psi}{\mathbb{X} \vDash \neg\diamond\neg\psi}\ \mathsf{Det2}$$

Proof. For any $c_0 \colon X \to S$ and $x \in X$ it holds:

$$\begin{aligned} \mathbb{X}, c_0, x \vDash \boxtimes\psi &\text{ iff } \mathbb{X}, G_{\mathbb{X}}(c_0), x \vDash \psi \\ &\text{iff } \mathbb{X}, G_{\mathbb{X}}(c_0), x \nvDash \neg\psi \\ &\text{iff } \mathbb{X}, c_0, x \nvDash \boxtimes\neg\psi \\ &\text{iff } \mathbb{X}, c_0, x \vDash \neg\boxtimes\neg\psi. \end{aligned}$$

(Note that, by definitions of the semantics of $\boxtimes$, $(\mathbb{X}, G_{\mathbb{X}}(c_0), x \vDash \neg\psi)$ and $(\mathbb{X}, c_0, x \vDash \boxtimes\neg\psi)$ have the same truth value).

The case where $\diamond = \diamond_m$ for some $m \in \mathscr{M}$ is analogous (but changing x to $m \circledast x$ instead of c_0 to $G_{\mathbb{X}}(c_0)$. □

Lemma 6.26: Generalisation

The following rule is sound:

$$\frac{\mathbb{X} \vDash \psi \quad \psi \in \Psi}{\mathbb{X} \vDash \diamond\psi}\,\mathsf{Gen}$$

Proof. In case $\diamond = \boxtimes$, take any $c_0 \colon X \to S$ and $x \in X$. We need to show that $\mathbb{X}, c_0, x \vDash \boxtimes\psi$, i.e., that $\mathbb{X}, G_{\mathbb{X}}(c_0), x \vDash \psi$. But $G_{\mathbb{X}}(c_0)$ is just a configuration, so this follows from $\mathbb{X} \vDash \psi$. □

Lemma 6.27: K-axiom

The following rule is sound:

$$\frac{s_0, s_1 \in S}{\mathbb{X} \vDash \diamond(\mathsf{v}_{s_0} \to \mathsf{v}_{s_1}) \to ((\diamond\mathsf{v}_{s_0}) \to (\diamond\mathsf{v}_{s_1}))}\,\mathsf{K}$$

Proof. We first consider the case where $\diamond = \diamond_m$ for some $m \in \mathscr{M}$. Assume any $s_0, s_1 \in S$, $c_0 \colon X \to S$ and $x \in X$. Applying Lemma 6.23, we assume that $\mathbb{X}, c_0, x \vDash \diamond_m(\mathsf{v}_{s_0} \to vo)$ and we need to show that $\mathbb{X}, c_0, x \vDash (\diamond_m\mathsf{v}_{s_0}) \to (\diamond_m\mathsf{v}_{s_1})$. We apply Lemma 6.23 again, and assume that $\mathbb{X}, c_0, x \vDash \diamond_m\mathsf{v}_{s_0}$; it remains to show that $\mathbb{X}, c_0, x \vDash \diamond_m\mathsf{v}_{s_1}$. Our two assumptions are equivalent to:

$$\mathbb{X}, c_0, m \circledast x \vDash \mathsf{v}_{s_0} \to \mathsf{v}_{s_1}$$

and

$$\mathbb{X}, c_0, m \circledast x \vDash \mathsf{v}_{s_0}.$$

respectively. We can apply MP on these latter forms to obtain $\mathbb{X}, c_0, m \circledast x \vDash \mathsf{v}_{s_1}$. This is equivalent to $\mathbb{X}, c_0, x \vDash \diamond_m\mathsf{v}_{s_1}$, which was to be shown.

The case $\diamond = \boxtimes$ is analogous, except that we replace c_0 by $G_{\mathbb{X}}(c_0)$ instead of x by $m \circledast x$. □

Space-time independence Suppose it is 11:55 AM. Then you could (1) walk downstairs and wait there for 5 minutes, and (2) wait for 5 minutes and thereafter go downstairs. But both options make you end up on the ground floor at 12:00. The key insight is that the stairs lead to the same place at any point in time.

Lemma 6.28: Space-Time Independence (STI)

For any $m \in \mathscr{M}$ and $\psi \in \Psi$, the following rules are sound:

$$\frac{\mathbb{X} \vDash \Diamond_m \boxtimes \psi}{\mathbb{X} \vDash \boxtimes \Diamond_m \psi}\ \mathsf{STI}$$

Proof. Directly from the semantics:

$$\begin{aligned} \mathbb{X}, c_0, x \vDash \Diamond_m \boxtimes \psi \text{ iff } & \mathbb{X}, c_0, m \circledast x \vDash \boxtimes \psi \\ \text{iff } & \mathbb{X}, G_{\mathbb{X}}(c_0), m \circledast x \vDash \psi \\ \text{iff } & \mathbb{X}, G_{\mathbb{X}}(c_0), x \vDash \Diamond_m \psi \\ \text{iff } & \mathbb{X}, c_0, x \vDash \boxtimes \Diamond_m \psi. \end{aligned}$$

□

Lemma 6.29: Double Negation

For any $\psi \in \Psi$, the following rule is sound:

$$\frac{}{\mathbb{X} \vDash \neg\neg\psi \leftrightarrow \psi}\ \mathsf{DN}$$

Proof. This follows from the definition of the semantics: a formula in a model $(\mathbb{X}, c_0)$ at a cell $x \in X$ is either satisfied or not. Thus, if $\mathbb{X}, c_0, x \nvDash \phi$ is *not* true (for any $\phi \in \Phi$), then the only remaining possibility is that $\mathbb{X}, c_0, x \vDash \phi$. So we observe that $\mathbb{X}, c_0, x \vDash \neg\neg\psi$ iff $\mathbb{X}, c_0, x \nvDash \neg\psi$. Now $\mathbb{X}, c_0, x \vDash \neg\psi$ holds iff $\mathbb{X}, c_0, x \nvDash \psi$, thus $\mathbb{X}, c_0, x \nvDash \neg\psi$ holds iff $\mathbb{X}, c_0, x \vDash \psi$. □

6.5 Summary of basic modal logic

This section constructed the basic modal logic $\mathcal{L}_{\mathsf{stg}}$, a cellular automaton analogue of Kripke modal logics. The frames $\mathbb{X} = (\mathscr{M}, N, S, \mathcal{X})$ in this logic are CA coalgebras in $\mathbf{CCA}_{\mathsf{G}}$, together with the signature $(\mathscr{M}, N, S)$ on which they are defined. Models are frames paired with an initial configuration $c_0 \colon X \to S$, which define a trace $[c_0, c_1, c_2, \dots]$ that assigns a state to every cell at every point in time. These states $s \in S$ act as atomic formulas, which are true at a cell $x \in X$ in a configuration c_0 iff c_0 assigns state s to x (i.e., $c_0(x) = s$). Paired with the Boolean connectives $\wedge$, $\vee$, $\neg$ and $\rightarrow$, these atomic formulas form the basic building blocks of the language $\mathcal{L}_{\mathsf{stg}}$. But by encapsulating subformulas in space $\Diamond_m$ and time $\boxtimes$ diamonds, we allow to evaluate subformulas at cells $m \circledast x$ for $m \in \mathscr{M}$, as well as in a the next configuration c_1 in the trace. Diamonds can be nested, which allow to specify any reachable point in the CA's graph and any future configuration c_t.

Unlike in Kripke modal logics, the atomic formulas $s \in S$ cannot be freely substituted by other formulas. We introduced a global variable v_s for every $s \in S$ that can be freely substituted by *time-invariant* formulas, while preserving formula satisfaction. Luckily, many other reasoning tools familiar from Kripke modal logics, such as the K-axiom, modus ponens, generalisation of valid formulas and the validity of all Boolean tautologies do carry over to $\mathcal{L}_{\mathsf{stg}}$.

The language $\mathcal{L}_{\mathsf{stg}}$ can express certain properties of a CA, such as the local rule (provided that the neighbourhood N is finite). But it also has its limits. Every diamond connects a cell in a specific point in the trace to a single other specific cell in a specific point in the trace. This makes it impossible to express properties such as "*for all $m \in \mathscr{M}$, it holds that ...*" or "*for all c_t with $t \in \mathbb{N}$ it holds that ...*". The next section will extend the language with infinitary disjunctions and (derived) infinitary conjunctions, which can encode such quantifications and greatly increases the expressivity of the language.

7 Logic extension: infinitary disjunctions

The expressivity of $\mathcal{L}_{\mathsf{stg}}$ can significantly be improved after extending it with infinitary disjunctions, which are practically existential quantifiers.

7.1 Intuition and motivation

We first motivate the need of this extension with three examples. Thereafter we present the formal extensions to the grammar and semantics.

Example 7.1 (Expressing all spatio-temporal patterns in a trace)**.** Example 6.10 showed how $\mathcal{L}_{\mathsf{st}}$ can express that a given cell has a certain state in a certain point in time in a trace. With the help of a finite number of conjunctions we could describe any finite patterns this way. With infinitary conjunctions (which automatically are as the duals $\bigwedge = \neg \bigvee \neg$ of infinitary disjunctions) we can describe any spatio-temporal pattern in a trace of a rooted CA.

Example 7.2 (Expressing "forever after" and "eventually")**.** There are many examples of properties that can be expressed with the notions of *forever* and *eventually*. For example, a CA may have a "dead" state $s_d \in S$ that a cell cannot leave once entered. This state satisfies the statement "*if in state s_d, then forever after in state s_d*". Logically, "*forever after ψ holds*" means that $⧖^n \psi$ holds for all choices of $n \in \mathbb{N}$ repetitions of the time diamond. Thus out statement would be:

$$s_d \to \bigwedge_{n \in \mathbb{N}} ⧖^n s_d. \tag{52}$$

As for another example, some CA have dynamics such that every trace will converge to a fixpoint in which all cells have a "dead" (quiescent) state s_d. However, the exact number of timesteps before a cell dies may vary with the initial configuration chosen. We only know there exists a choice of a finite number of timesteps before this occurs, which is an infinite disjunction over all number of timesteps. Hence we can express "*eventually state s_d will be attained*" as:

$$\bigvee_{\ell \in \mathbb{N}} ⧖^\ell s_d. \tag{53}$$

This does not yet express that the cell will remain in s_q. However, combining these two examples, we can express "*every cell will eventually obtain state s_d, and once in s_d, it will retain this state forever after*":

$$\bigvee_{\ell \in \mathbb{N}} ⧖^\ell \left(\bigwedge_{n \in \mathbb{N}} ⧖^n s_d \right). \tag{54}$$

Example 7.3 (Quiescent states)**.** A *quiescent state* is a state $s_q \in S$ such that if a cell and all its neighbours have state s_q, then the cell will stay in state s_q the next timestep. Quiescent states are useful for simulating a CA with in infinite grid on a real computer: this is possible if the number of non-quiescent cells is finite, since the quiescent cells do not need to be stored or computed explicitly.

If we allow infinitary disjunctions, then we obtain infinitary conjunctions as their dual ($\bigwedge = \neg \bigvee \neg$). With these conjunctions we can express "*all neighbours have state s_q*" in a formula as $\bigwedge_{n \in N} \diamond_n s_q$. The statement "*the cell will stay in state s_q*" means that state s_q will occur the next timestep, i.e., the formula $⧖ s_q$. So quiescence of s_q can be expressed with

$$\left(\bigwedge_{n \in N} \diamond_n s_q \right) \to ⧖ s_q. \tag{55}$$

Finally, "*there exists a quiescent state*" holds as soon as (55) is valid for some choice of $s_q \in S$. This is essentially a disjunction over all states:

$$\bigvee_{s\in S}\left(\left(\bigwedge_{n\in N}\Diamond_n s\right)\to \boxtimes s\right). \tag{56}$$

7.2 Formal definitions of infinitary disjunctions

We extend the language $\mathcal{L}_{\mathsf{stg}}$ with a new constructor $\bigvee_{i\in I}\psi_i$ for every index set I and set of subformulas $\{\psi_i\}_{i\in I}$. We interpret $\bigvee_{i\in I}\psi_i$ the same way as the (possible infinite) formula obtained as the disjunction of ψ_i over all $i \in I$. While results in very large and very general language $\mathcal{L}'_{\mathsf{CA}}$, we will primarily continue to work with a subset $\mathcal{L}_{\mathsf{CA}} \subseteq \mathcal{L}'_{\mathsf{CA}}$ of a few specific choices of sets of formulas $\{\psi_i\}_{i\in I}$.

Definition 7.4: Grammar of $\mathcal{L}'_{\mathsf{CA}}$

The well-formed formulas Ψ that form the grammar of $\mathcal{L}'_{\mathsf{CA}}$ are given by:

$$\Psi ::= \bot \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \Diamond_m\psi \mid \boxtimes\psi \mid s \mid \mathsf{v}_s \mid \bigvee_{i\in I}\psi_i$$

where $\psi, \psi_1, \psi_2 \in \Psi$, $m \in \mathscr{M}$ and $\{\psi_i\}_{i\in I} \subseteq \Psi$.

Definition 7.5: Semantics of $\mathcal{L}'_{\mathsf{CA}}$

The semantics of $\mathcal{L}'_{\mathsf{CA}}$ are as for $\mathcal{L}_{\mathsf{stg}}$ (see Definition 6.9 and Eq. (49)), extended with the rule:

$$\mathbb{X}, G^t_{\mathbb{X}}(c_0), x \vDash \bigvee_{i\in I}\psi_i \text{ iff there exists an } i \in I \text{ such that } \mathbb{X}, G^t_{\mathbb{X}}(c_0), x \vDash \psi_i.$$

Remark 7.6. We will write $\bigwedge_{i\in I}\psi_i$ to denote $\neg\bigvee_{i\in I}\neg\psi_i$; observe we can deduce the familiar semantics:

$$\mathbb{X}, G^t_{\mathbb{X}}(c_0), x \vDash \bigvee_{i\in I}\psi_i \text{ iff for all } i \in I \text{ it holds that } \mathbb{X}, G^t_{\mathbb{X}}(c_0), x \vDash \psi_i.$$

Remark 7.7. All the deduction rules of Section 6.4 are also sound for $\mathcal{L}'_{\mathsf{CA}}$. This is straightforward to see: the given proofs also work for $\mathcal{L}'_{\mathsf{CA}}$.

Remark 7.8. Uniform substitution of time-invariant variables (Lemma 6.16 and Lemma 6.17) also holds for $\mathcal{L}'_{\mathsf{CA}}$. The only required modification in the proofs is adding the $\theta = \bigvee_{i\in I}\psi_i$ case to the structural induction on θ in Lemma 6.17 (with the induction hypothesis assumed for each ψ_i). However, this case is analogous to the binary disjunction $\theta = \psi_1 \vee \psi_2$ case since we have a similar distribution of the substitution: $\left(\bigvee_{i\in I}\psi_i\right)\left[{}^{\phi}/_{\mathsf{v}_s}\right] = \bigvee_{i\in I}(\psi_i\left[{}^{\phi}/_{\mathsf{v}_s}\right])$.

We will be working with $\mathcal{L}_{\mathsf{CA}}$, which uses the following infinitary disjunctions:

Definition 7.9: Subset $\mathcal{L}_{\mathsf{CA}}$ of $\mathcal{L}'_{\mathsf{CA}}$

$\mathcal{L}_{\mathsf{CA}}$ is $\mathcal{L}'_{\mathsf{CA}}$ but with restricted choices of the sets of subformulas used in infinitary disjunctions.

The first few of those sets are "decorated" version of a fixed formula $\psi \in \Psi$, and defined for each chose of ψ. Let $A \overset{\text{def}}{=} \{\mathscr{M}, \mathscr{M}\setminus\mathbf{1}, N, N\setminus\mathbf{1}\}$, then the sets of decorations for ψ over signature $(\mathscr{M}, N, S)$ in $\mathcal{L}_{\mathsf{CA}}$ are:

- The set of steps in an arbitrary direction:

$$\{\diamond_m \,|\, m \in Z\} \tag{57}$$

  for any choice of $Z \in A$. Disjunctions over this set will be denoted as $\bigvee_{m\in Z} \diamond_m \psi$.

- The set of paths over Z for any choice of $Z \in A$ and $\psi \in \Psi$:

$$\mathsf{Paths}(Z) \stackrel{\text{def}}{=} \{\diamond_{m_1} \diamond_{m_2} \ldots \diamond_{m_k} \,|\, m_i \in Z, k \in \mathbb{N}\}. \tag{58}$$

  Notation: $\bigvee_{\pi\in\mathsf{Paths}(Z)} \pi\psi$.

- For any choice of Z and $m \in Z$ the set of repetitions of $\diamond_m$:

$$\{\diamond_m^k \,|\, k \in \mathbb{N}\}. \tag{59}$$

  Notation: $\bigvee_{k\in\mathbb{N}} \diamond_m^k \psi$.

- The set of repetitions of the time modality:

$$\{\mathbb{x}^k \,|\, k \in \mathbb{N}\}. \tag{60}$$

  Notation: $\bigvee_{k\in\mathbb{N}} \mathbb{x}^k \psi$.

$\mathcal{L}_{\mathsf{CA}}$ may also includes disjunctions over set of states $s \in S$ and the set of global variables v_s with $s \in S$. This are sets of atomic formulas (and do not depend on ψ).

Finally, $\mathcal{L}_{\mathsf{CA}}$ also has all variants of the following *until modality*:

$$\phi \mathsf{U} \psi \stackrel{\text{def}}{=} \bigvee k \in \mathbb{N}(\mathbb{x}^k \psi) \wedge \bigwedge_{0\leq j\leq k-1} \mathbb{x}^j \psi. \tag{61}$$

(Intuitively: "*ψ will eventually hold after k timesteps, and if $k > 0$ then ϕ holds after any $0, 1, \ldots, k-1$ timesteps.*")

Remark 7.10 (Cardinality of formulas in $\mathcal{L}_{\mathsf{CA}}$)**.** The cardinality on the size of a set of formulas in infinitary disjunctions in $\mathcal{L}_{\mathsf{CA}}$ is bounded by the cardinality κ of the signature $(\mathscr{M}, N, S)$. The effective length of formulas (when unfolding the infinitary disjunctions into actual infinite sequences of disjunctions) can be any finite power of κ since we can nest infinitary disjunctions (e.g., $\bigvee_{m\in\mathscr{M}} \bigvee_{n\in\mathscr{M}} \bigvee_{\ell\in\mathscr{M}} \diamond_m \diamond_n \diamond_\ell \psi$ is effectively a disjunction over $\mathsf{card}(\mathscr{M})^3$ subformulas).

Remark 7.11 (Until modality)**.** Many variants of the "until modality" are possible: e.g. one could analogously define it for repetitions of a space diamonds $\diamond_m$, or drop the constraint that ψ must eventually hold (then it is possible that ϕ holds forever and ψ never).

Also interesting is to define the following "until modality" for paths. Paths work a bit different than mere repetition, so we define $\phi \mathsf{U}_{\mathsf{Paths}(Z)} \psi$ to denote that every path first visits a sequence of cells where ϕ hold, then a cell where ψ holds, and continues without further constraints. Since we do not both control the length and direction of the path, it may keep circling in the ϕ-region; the solution is simply to use the "until"-variant where the end condition ψ is not guaranteed to hold.

Formally, let $\sqsubset$ denote the is-a-strict-prefix relation (e.g., $\diamond_{m_1} \diamond_{m_2} \diamond_{m_3} \sqsubset \diamond_{m_1} \diamond_{m_2} \diamond_{m_3} \diamond_{m_4}$), then we can define:

$$\phi \mathsf{U}_{\mathsf{Paths}(Z)} \psi \stackrel{\text{def}}{=} \bigwedge_{\pi\in\mathsf{Paths}\, Z} \left(\left(\pi\phi \wedge \bigwedge_{\theta\sqsubset\pi} \theta\phi \right) \vee \bigvee_{\theta\sqsubset\pi} \left(\theta\psi \wedge \bigwedge_{\omega\sqsubset\theta} \omega\phi \right) \right).$$

The interpretation is easiest to explain when we take the CA on the grid $\mathbb{Z}^2$ with two colours $S = \{\circ, \bullet\}$ and pick $\phi := \bullet$ and $\psi := \circ$. Then $\pi \circ \wedge \bigwedge_{\theta \sqsubset \pi} \theta \bullet$ means "*all cells on the path* π *are* $\circ$", and $\bigvee_{\theta \sqsubset \pi} \left(\theta \bullet \wedge \bigwedge_{\omega \sqsubset \theta} \omega \circ\right)$ means "*there exists a prefix of* π *that first only visits* $\circ$ *cells and ends in a cell in state* $\bullet$." Thus $\circ \mathsf{U}_{\mathsf{Paths}(\mathbb{Z}^2)} \bullet$ reads "*I am in a blob of* $\circ$ *cells, and this blob is surrounded by* $\bullet$ *cells.*" See Figure 19 for a visualisation.

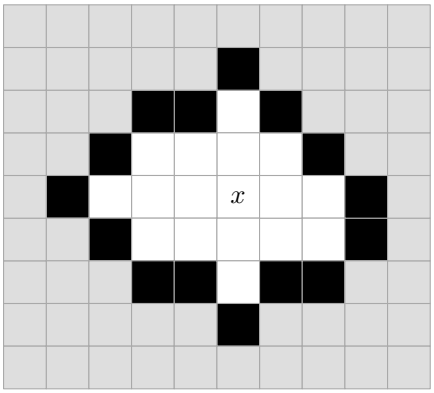


Figure 19: Example configuration of black and white states ($\{\circ, \bullet\}$) on the grid $\mathbb{Z}^2$, the cell x satisfies $\circ \mathsf{U}_{\mathsf{Paths}(\mathbb{Z}^2)} \bullet$. The formula does not concern the states of gray cells, which can therefore be arbitrary.

7.3 Examples of expressible properties

The expressivity of $\mathcal{L}_{\mathsf{CA}}$ is best illustrated by concrete examples. In particular, we will explore certain formulas that are valid on a frame iff that frame has a specific property.

Most examples only apply to *rooted* (recall from Definition 3.8) frames with at least two states ($\mathsf{card}(S) \geq 2$). We will refer to these CA as *nice* CA.

Rootedness ensures that there is a cell that can describe properties of every other cell, i.e., global properties. At least two states are needed because a "CA" with $\mathsf{card}(S) = 1$ has no dynamics at all, and the only non$-\bot$ atomic formulas are states; hence all cells satisfy the same formulas at any timestep and cannot be told apart.

Local rules Lemma 6.12 showed how to express local rules for finite neighbourhoods N as formulas; with infitary conjunctions, the same proof generalises to local rules on infinite N.

Eventually everyone We can express that, after a given number of timesteps, all cells satisfy a given formula ψ.

Lemma 7.12

A nice CA $\mathbb{X}$ validates

$$\mathsf{EvEv}(\psi) \stackrel{\text{def}}{=} \bigvee_{k \in \mathbb{N}} \mathbb{x}^k \bigwedge_{m \in \mathscr{M}} \Diamond_m \psi \tag{62}$$

iff for every initial configuration $c_0 \colon X \to S$ there exists a $t \in \mathbb{N}$ s.t. $\mathbb{X}, G^t_{\mathbb{X}}, x \vDash \psi$ for all $x \in X$.

Proof. Let $r \in X$ be the root. Then in a given model c_0 we have that $\mathbb{X}, c_0, r \vDash \mathsf{EvEv}(\psi)$ iff there exists a $t \in \mathbb{N}$ s.t. $\mathbb{X}, G^t_{\mathbb{X}}(c_0), r \vDash \bigwedge_{m \in \mathscr{M}} \Diamond_m \psi$. The latter holds iff $\mathbb{X}, G^t_{\mathbb{X}}(c_0), m \otimes r \vDash \psi$ for all $m \in \mathscr{M}$. Since r is the root, $\mathscr{M} \otimes r = X$, thus this holds iff all cells $x \in X$ satisfy ψ at time t. □

Example 7.13 (Eventually everyone is dead)**.** Suppose we have a "dead" state $s_d \in S$. Then $\bigwedge_{t\in\mathbb{N}} \mathbb{x}^t s_d$ expresses that a cell is dead and stays dead forever after. Hence

$$\mathsf{EvEv}(\bigwedge_{t\in\mathbb{N}} \mathbb{x}^t s_d) = \bigvee_{k\in\mathbb{N}} \mathbb{x}^k \bigwedge_{m\in\mathscr{M}} \Diamond_m \bigwedge_{t\in\mathbb{N}} \mathbb{x}^t s_d \tag{63}$$

expresses (in a nice CA) that "*eventually, every cell will be forever dead*".

The trace will become periodic A CA $\mathbb{X} = (\mathscr{M}, N, S, \mathcal{X})$ might be such that for every initial configuration $c_0 \colon X \to S$, the resulting trace $[c_0, c_1, c_2, c_3, \dots]$ (computed c.f. (28)) becomes periodic. More precisely, there exists (for every c_0) a starting timesteps $k \in \mathbb{N}$ and a period $p \in \mathbb{N}$ such that $c_{k+i} = c_{k+i+p} = c_{k+i+2p} = c_{k+i+3p} = \dots$ for all $i \in \mathbb{N}$. An intuitive example is a traffic light: it is a periodic 3-cell CA with states $\{\text{off}, \text{green}, \text{orange}, \text{red}\}$.

Lemma 7.14

A nice CA $\mathbb{X}$ validates

$$\bigvee_{p\in\mathbb{N}} \mathsf{EvEv}\left(\psi(p)\right) \tag{64}$$

where

$$\psi(p) \stackrel{\text{def}}{=} \bigwedge_{k\in\mathbb{N}} \bigwedge_{s\in S} \mathbb{x}^k(s \to \mathbb{x}^p s) \tag{65}$$

iff it all its traces eventually become periodic.

Proof. A trace is periodic with period p iff every cell has a sequence of states that repeats with period p. Thus, by Lemma 7.12, it suffices to show that $\psi(p)$ expresses "*my states are repeating periodically with period p*".

Assume an arbitrary initial configuration $c_0 \colon X \to S$ and write c_i for $G^i_{\mathbb{X}}(c_0)$. We need to show that $\mathbb{X}, c_i, x \vDash \psi(p)$ iff x's states in the suffix $[c_i, c_{i+1}, c_{i+2}, \dots]$ of the trace repeat with period p.

For the first direction, assume that

$$\mathbb{X}, c_i, x \vDash \psi(p). \tag{66}$$

Take an arbitrary point $0 \le z \le p-1$ in the period and let $s_0 := c_{i+z}(x)$, then we need to show that $\mathbb{X}, c_{i+z+np}, x \vDash s_0$ for all $n \in \mathbb{N}$. We proceed with induction on n. Note that the base case holds trivially by definition of s_0.

For the inductive step, assume the induction hypothesis $\mathbb{X}, c_{i+z+np}, x \vDash s_0$. If we choose $k := z + np$ and $s := s_0$ in (66) we obtain $\mathbb{X}, c_{i+z+pn}, x \vDash s_0 \to \mathbb{x}^p s_0$. Using the inductive hypothesis and modus ponens (Lemma 6.21) we find $\mathbb{X}, c_{i+z+pn}, x \vDash \mathbb{x}^p s_0$. This holds iff $\mathbb{X}, c_{i+z+p(n+1)}, x \vDash s_0$, which was the goal of the inductive step.

For the reverse direction, assume $c_{i+z} = c_{i+z+np}$ for all $0 \le z \le p-1$ and $n \in \mathbb{N}$. It is to show that $\mathbb{X}, c_i, x \vDash \bigwedge_{k\in\mathbb{N}} \bigwedge_{s\in S} \mathbb{x}^k(s \to \mathbb{x}^p s)$. It suffices to show that, for an arbitrary $k \in \mathbb{N}$ and $s_0 \in S$ it holds that $\mathbb{X}, c_{i+k}, x \vDash s_0 \to \mathbb{x}^p s_0$. We use the method from propositional logic of "$\to$"-introduction (Lemma 6.23). So assume $\mathbb{X}, c_{i+k}, x \vDash s_0$; this holds iff $c_{i+k}(x) = s_0$. But by our periodicity assumption we know $c_{i+k} = c_{i+k+p}$, so $c_{i+k+p}(x) = s_0$ as well. The latter is equivalent to $\mathbb{X}, c_{i+k+p}, x \vDash s_0$, which holds iff $\mathbb{X}, c_{i+k+p}, x, \vDash \mathbb{x}^p s_0$. □

Undecidability Two of the previous examples are actually already show that $\mathcal{L}_{\mathsf{CA}}$ can encode undecidable properties of CA. On the one hand, this highlights the expressivity of the language; on the other hand, it seems not practically possible to compute whether a formula is valid on a given CA.

Corollary 7.15

Frame validity of a formula in $\mathcal{L}_{\mathsf{CA}}$ is undecidable. (I.e., there exist a frame $\mathbb{X}$ and a formula ψ in $\mathcal{L}_{\mathsf{CA}}$ such that it is undecidable whether $\mathbb{X} \vDash \psi$).

Proof. We give two proofs that use a different classical result.

A CA $\mathbb{X}$ is *nilpotent* if there exists a $t \in \mathbb{N}$ and a configuration $c_d \colon X \to S$ such that c_d is a fixpoint of $G_{\mathbb{X}}$, and $G^t_{\mathbb{X}}(c_0) = c_q$ for any initial configuration $c_0 \colon X \to S$. Kari proved in 1992 [19] that it is undecidable for a CA on $\mathbb{Z}$ whether it is nilpotent. This cited paper also shows that in a nilpotent CA, c_d necessarily assigns all cells the same state, say $s_d \in S$. Example 7.13 exactly expresses that the configuration const_{s_d} will be reached; we only need to add "*there exists a $s_d \in S$ such that eventually all cells will be in state s_d*". Hence a CA is nilpotent iff it validates:

$$\bigvee_{s_d \in S} \mathsf{EvEv}\left(\bigwedge_{k\in\mathbb{N}} \mathbb{X}^k s_d\right) \overset{\text{def}}{=} \bigvee_{s_d\in S}\bigvee_{t\in\mathbb{N}} \mathbb{X}^t\left(\bigwedge_{m\in\mathscr{M}}\left(\Diamond_m \bigwedge_{k\in\mathbb{N}} \mathbb{X}^k s_d\right)\right).$$

A CA is *periodic* if there exists an $p \in \mathbb{N}$ such that $G^p_{\mathbb{X}} = \mathsf{Id}$. Since the trace are just the outputs of $G^t_{\mathbb{X}}(c_0)$ for all t, this definition is equivalent to periodicity of the trace. Lemma 7.14 shows how to express that the trace will *eventually* become periodic. Removing the disjunction $\bigvee_{k\in\mathbb{N}} \mathbb{X}^k$ over the starting point in time of the periodicity (it should start immediately), we obtain the formula that expresses periodicity:

$$\bigvee_{p\in\mathbb{N}}\bigwedge_{m\in\mathscr{M}} \Diamond_m\left(\bigwedge_{t\in\mathbb{N}}\bigwedge_{s\in S} \mathbb{X}^t(s \to \mathbb{X}^p s)\right).$$

Whether a CA on $\mathbb{Z}$ is periodic is undecidable (proven by Kari and Ollinger in 2008 [21]).

Note that the classical undecidability results remain when restricting to nice CA. $\mathbb{Z}$ is a rooted grid, and both problems are trivial to decide when a CA has only a single state (hence it is not possible that only the subclass of CA on $\mathbb{Z}$ with $\mathsf{card}(S) = 1$ is undecidable). □

Steady states A cell x may, in a given model $(\mathbb{X}, c_0)$, eventually reach a steady state. That is, $c_i(x) = c_{i+k}(x)$ for some $i \in \mathbb{N}$ and all $k \in \mathbb{N}$. This is simply expressed as $\mathbb{X}, c_0, x \vDash \bigvee_{i\in\mathbb{N}}\bigvee_{s\in S} \mathbb{X}^i\left(\bigwedge_{k\in\mathbb{N}} \mathbb{X}^k s\right)$. More interestingly, we can also show that a cell (or even stronger, no cell) will never reach a steady state.

Lemma 7.16: Never reach a steady state

A cell $x \in X$ will never reach a steady state in model $(\mathbb{X}, c_0)$ iff

$$\mathbb{X}, c_0, x \vDash \bigwedge_{s\in S}\left(s \to \bigvee_{i\in\mathbb{N}} \mathbb{X}^i \neg s\right). \tag{67}$$

It follows that no cell in any model on a nice CA will reach a steady state iff

$$\mathbb{X} \vDash \bigwedge_{m\in\mathscr{M}}\bigwedge_{s\in S}\left(s \to \bigvee_{i\in\mathbb{N}} \mathbb{X}^i \neg s\right). \tag{68}$$

Proof sketch. A cell x will reach a steady iff it will reach some state s s.t. all future states of x are also s. Equivalently, a state is not steady iff there exist a future state of x that is different. Thus it suffices to show that x has no steady state iff every state will eventually be succeeded by another state, which is exactly expressed by (67). □

The local rule of a CA may ignore the state of a given neighbour at path $n \in N$. I.e., $\gamma_x(z) = \gamma_x(z')$ for all $z, z' \colon \mathscr{M} \mathbin{⨂} x \to S$ that only differ on input $n \mathbin{⨂} x$.

Lemma 7.17

Suppose $N = \{n_0, n_1, \ldots, n_k\}$ is a finite neighbourhood. The local rule γ of a frame $\mathbb{X}$ always ignores neighbour $n_0 \in N$ iff

$$\mathbb{X} \vDash \bigwedge_{s_0 \in S} \bigwedge_{s_1 \in S} \cdots \bigwedge_{s_k \in S} \bigwedge_{s' \in S} ([((\Diamond_{n_0} s_0) \wedge (\Diamond_{n_1} s_1) \wedge \cdots \wedge (\Diamond_{n_k} s_k)) \to ⧖ s'] \to [((\Diamond_{n_1} s_1) \wedge \cdots \wedge (\Diamond_{n_k} s_k)) \to ⧖ s']) \tag{69}$$

Information flow graph In configuration c_i in a trace of a model $(\mathbb{X}, c_i)$, the state of a cell x in the next timestep (i.e., $c_{i+1}(x)$) is entirely determined by (1) γ_x and (2) the states of x's neighbours in c_i. But the neighbours' states depend on the states of their neighbours in c_{i-1}, which in turn depend on their neighbours' states in c_{i-2}, and so on, until c_0. Thus, the transitive closure of the is-a-neighbour-of relation relates a cell x to all the cells x' that may influence the states occurring at x in the trace. If x is not related to a cell x'' this way, then no information coming from x'' can influence x. Hence, we can construct the graph that depicts how information can "flow" in a CA:

Definition 7.18: Information flow graph

Given a frame $\mathbb{X} = (\mathscr{M}, N, S, \mathcal{X})$, define the *information flow graph* $\mathfrak{G}^N = (V, E)$ to be the directed graph where:

- The vertices are the cells: $V = X$.
- The edges $E \subseteq V \times V$ are the is-a-neighbour-of-relation: $(x, x') \in E$ iff $x' \in N \mathbin{⨂} x$.

In many cases, $\mathfrak{G}^N$ has the same shape as "the grid" (which is the graph obtained by $(x, x') \in E$ iff $x' \in \mathscr{M} \mathbin{⨂} x$), such as for the forktree (Figure 5) and the $\mathbb{Z}^2$ grid in CGOL. An interesting exception is the CA $\mathbb{Y}$ of Example 8.28: this one has an hexagonal information graph.

There are many important properties of $\mathfrak{G}^N$ that can be expressed as $\mathcal{L}_{\mathsf{CA}}$ formulas. A first example is essential undirectedness of $\mathfrak{G}^N$:

Definition 7.19: Essential undirected graph

A graph (V, E) is *essentially undirected* if E is a symmetric relation (i.e., $(x, x') \in E$ iff $(x', x) \in E$).

Lemma 7.20

The information graph $\mathfrak{G}^N$ of a nice CA $\mathbb{X}$ is essentially undirected iff

$$\mathbb{X} \vDash \mathsf{v}_{s_0} \to \bigwedge_{m \in N} \bigvee_{n \in N} \Diamond_m \Diamond_n \mathsf{v}_{s_0} \tag{70}$$

where $s_0 \neq s_1 \in S$ are distinct states.

Proof. Suppose $\mathbb{X}$ validates (70) and $(x, x') \in E$. Choose states $s_0, s_1 \in S$ such that $s_0 \neq s_1$, and pick $c_0 \colon X \to S_X$ as $c_0(x'') = \begin{cases} s_0 & x'' = x \\ s_1 & \text{otherwise} \end{cases}$. Then $\mathbb{X}, c_0, x \vDash \mathsf{v}_{s_0}$, and by modus ponens we find that $\mathbb{X}, c_0, x \vDash$

$\bigwedge_{m\in N}\bigvee_{n\in N}\Diamond_m\Diamond_n \mathsf{v}_{s_0}$. Since $(x,x')\in E$, there exists an $m\in N$ such that $x'=m\mathbin{⨀}x$. Using this m, by $\wedge$-elimination we find $\mathbb{X},c_0,x\vDash\bigvee_{n\in N}\Diamond_m\Diamond_n\mathsf{v}_{s_0}$. Thus there exists an $n\in N$ such that $\mathbb{X},c_0,x\vDash\Diamond_m\Diamond_n\mathsf{v}_{s_0}$, which holds iff $\mathbb{X},c_0,x'\vDash\Diamond_n\mathsf{v}_{s_0}$, which wholds iff $\mathbb{X},c_0,n\mathbin{⨀}x'\vDash\mathsf{v}_{s_0}$. So x' has a n-neighbour with state s_0; but x is the only cell with state s_0 in c_0, so $n\mathbin{⨀}x'=x$. Hence n witnesses that $(x',x)\in E$ as well.

For the other direction, suppose E is symmetric and take any $x\in X$ and $c_0\colon X\to S_X$. By Lemma 6.23 it suffices to assume $\mathbb{X},c_0,x\vDash\mathsf{v}_{s_0}$ and to show $\mathbb{X},c_0,x\vDash\bigwedge_{m\in N}\bigvee_{n\in N}\Diamond_m\Diamond_n\mathsf{v}_{s_0}$. So take an arbitrary $m\in N$. By definition $(x,m\mathbin{⨀}x)\in E$, and since E is symmetric, this means $(m\mathbin{⨀}x,x)\in E$. The latter implies the existence of an $n\in N$ s.t. $n\mathbin{⨀}m\mathbin{⨀}x=x$. Rewriting this into $\mathbb{X},c_0,x\vDash\mathsf{v}_{s_0}$ gives $\mathbb{X},c_0,n\mathbin{⨀}m\mathbin{⨀}x\vDash\mathsf{v}_{s_0}$, i.e., $\mathbb{X},c_0,x\vDash\Diamond_m\Diamond_n\mathsf{v}_{s_0}$, which implies $\mathbb{X},c_0,x\vDash\bigvee_{n\in N}\Diamond_m\Diamond_n\mathsf{v}_{s_0}$. As m was arbitrary, we obtain the outer the infinitary conjunction. □

Other interesting properties of $\mathfrak{G}^N$ in $\mathcal{L}_{\mathsf{CA}}$ as well. The proofs are similar as the essential undirectedness formula, and therefore omitted.

Commuting information flow graph Call a $\mathfrak{G}^N$ *commuting* if first following the step m and then n leads to the same cell as first following n and then m.

Lemma 7.21

The information graph $\mathfrak{G}^N$ of a nice CA $\mathbb{X}$ is *commuting* iff

$$\mathbb{X}\vDash\bigwedge_{m\in N}\bigwedge_{n\in N}\left(\Diamond_m\Diamond_n\mathsf{v}_{s_{s_0}}\leftrightarrow\Diamond_n\Diamond_m\mathsf{v}_{s_{s_0}}\right). \tag{71}$$

Many regular grids such as $\mathbb{Z}^2$ are commuting. In fact, $\mathfrak{G}^N$ will commute as soon as $\mathscr{M}$ is a group; this ensures that $\bullet$ is commuting and from Lemma 6.5) it follows that $\Diamond_m\Diamond_n=\Diamond_{n\bullet m}=\Diamond_{m\bullet n}=\Diamond_n\Diamond_m$.

Reachability implies dependence It is interesting to know whether the existence of a path $m\in\mathscr{M}$ from x to a cell $m\mathbin{⨀}x$ implies that information in $m\mathbin{⨀}x$ can flow to x. That is, $(x,m\mathbin{⨀}x)\in E$ for all $m\in\mathscr{M}$. Many CA have this property, but for example not in $\mathbb{Z}$ when using the neighbourhood $N=\{-1,0\}$; then the state of x is never influenced by a past state of $x+1$.

Lemma 7.22

The information graph $\mathfrak{G}^N$ of nice CA $\mathbb{X}$ has the property $(x,m\mathbin{⨀}x)\in E$ for all $x\in X$ and $m\in\mathbin{⨀}$ iff

$$\mathbb{X}\vDash\bigvee_{m\in\mathscr{M}}\left((\Diamond_m\mathsf{v}_{s_0})\to\bigvee_{\pi\in\mathsf{Paths}(N)}\pi\mathsf{v}_{s_0}\right). \tag{72}$$

One-way Cellular Automata A One-way Cellular Automaton (OCA) is a CA s.t. information can "flow in only one direction". The typical example is $\mathbb{Z}$ with $N=-1,0$; in such a CA, cells do not depend on their right neighbour, thus information can only flow from left to right. There are different ways to generalise one-wayness to other CA, but the following expresses it in terms of $\mathfrak{G}^N$:

Definition 7.23: One-way Cellular Automaton (OCA)

A *One-way Cellular Automaton* is a CA whose information graph $\mathfrak{G}^N$ has no cycles except for self-loops $(x,x)\in E$.

The property "*is an OCA*" cannot be expressed via validity of a $\mathcal{L}'_{\mathsf{CA}}$ formula; we will prove this in Lemma 8.30 using the results of Section 8. However, $\mathcal{L}_{\mathsf{CA}}$ can express that a CA is *not* an OCA.

Lemma 7.24

A nice CA $\mathbb{X}$ is *not* an OCA if

$$\mathbb{X} \vDash \mathsf{v}_{s_0} \to \bigvee_{\pi \in \mathsf{Paths}(N \setminus \{\mathbf{1}\})} \pi \mathsf{v}_{s_0}. \tag{73}$$

This is an "*if*" statement, and not an "*iff*"; indeed, if $\mathbb{X}$ does not validate (73), then that only means that a counterexample exists. We have such a counterexample as soon as there exists *some* cell that does not lie on a nontrivial cycle in $\mathfrak{G}^N$; it does not imply that *no* cell has such a nontrivial cycle!

7.4 Summary of infinitary disjunctions

This section showed that $\mathcal{L}_{\mathsf{stg}}$ can be extended with disjunctions over arbitrary (and possibly infinite) sets to obtain the logical language $\mathcal{L}'_{\mathsf{CA}}$. The fragment $\mathcal{L}_{\mathsf{CA}}$, which only takes disjunctions over sets directly in or related to the sets in a frame $(\mathscr{M}, N, S, \mathcal{X})$, can already express many properties of CA that $\mathcal{L}_{\mathsf{stg}}$ cannot. This includes properties of the global dynamics, such as the property of the trace converging to a fixpoint with all cells in a permanent "dead" state.

8 Correspondence semantical and logical bisimulation

There are two ways to compare two cells $x \in \mathcal{X}$ and $y \in \mathcal{Y}$: we can compare the sets $\mathsf{Log}(x)$ and $\mathsf{Log}(y)$ of logical formulas they validate, or we can directly compare them categorically as elements of coalgebras in $\mathbf{CCA}_{\mathsf{G}}$. This situation also occurs in modal logics for Kripke frames: the Hennessy-Milner theorem [12] states that points $x \in \mathbb{F}$ and $y \in \mathbb{G}$ are semantically similar (*bisimilar*, denoted $x \leftrightarroweq y$) iff they are logically equivalent ($x \leftrightsquigarrow y$) (provided that the frames are *image finite*: every point has only finitely many successors for each given edge colour).

Since the semantics of the logic are entirely defined in the dynamics of a coalgebra, it is clear that similar coalgebras should indeed validate a similar logic. Conversely, the logic $\mathsf{Log}(x)$ of a cell x in a CA frame $\mathbb{X} = (\mathscr{M}, N, S, X, \langle \chi, \gamma \rangle)$ can express the essential data of the dynamics at x, such as the local rule γ_x (Lemma 6.12) and the neighbourhood layout N_x (Example 6.8), thus we expect a similar logic to imply similar cells.

This section will first develop an intuition for the right CA definitions of *similarity* and *logical equivalence* with a few examples. We first show an example of similarity between CA with the same monoid and state set; Section 8.2 will prove a HM theorem for this setting. Thereafter we illustrate that a more general notion of similarity is needed to capture many pairs of CA that can directly simulate each other. The solution is to define a general notion of simulation, which is roughly a coalgebra homomorphism extended with a monoid homomorphism and a state set mapping. We also need a corresponding logical embedding that translates occurrences of monoid elements and states in formulas. A HM theorem proving that a cell x can be simulated by y iff the logic of x can be embedded in y can be found in Section 8.3. Note that this implies that x and y can mutually simulate each other (*mutual similarity*, our generalisation of *similarity*) iff their logics can be embedded into each other (*mutual logical simulation*). Example 8.28 concludes this section with a less trivial mutual similarity from the literature.

The results in this section apply to all the languages $\mathcal{L}_{\mathsf{stg}}$, $\mathcal{L}'_{\mathsf{CA}}$ and $\mathcal{L}_{\mathsf{CA}}$. However, for $\mathcal{L}_{\mathsf{stg}}$ there often is the additional condition that the neighbourhood N is finite (the apparent analog to image-finiteness condition of the Kripke logic HM theorem).

8.1 Examples of similarity

The first example shows similarity between CA with the same monoid and state set. This specific form of similarity can be directly phrased in terms of coalgebra homomorphisms.

8.1.1 A basic example of similarity

A starting point is to define logical equivalence $x \leftrightsquigarrow y$ simply as $\mathsf{Log}(x) = \mathsf{Log}(y)$, for cells $x \in \mathcal{X} = (X, \langle \chi, \gamma \rangle)$ and $y \in \mathcal{Y} = (Y, \langle \xi, \delta \rangle)$. As for coalgebraic similarity $x \leftrightarrows y$, called *similarity*, we should demands that $N_x = N_y$ (similar neighbourhood layout) and that $\gamma_x = \delta_y$ (same dynamics). This demand is satisfied if there exist homomorphisms

$$\overrightarrow{\mathscr{M} \circledS x} \underset{k}{\overset{h}{\rightleftarrows}} \overrightarrow{\mathscr{M} \circledS y}$$

such that $h(x) = y$ and $k(y) = x$, because (1) h and k are surjective on $\mathscr{M} \circledS y$ and $\mathscr{M} \circledS x$ respectively (Lemma 5.4), which already implies $N_x = N_y$, and (2) Theorem 5.9 guarantees that $\gamma_x = \delta_y$.

Example 8.1. Take the binary tree as monoid, i.e., the monoid $\mathscr{T} = (\{L, R\}^*, \bullet, \varepsilon)$ freely generated over $\{L, R\}$, and the neighbourhood $N = \{\varepsilon, L, R\}$ (i.e., the node itself and its two children). Now consider a frame $\mathbb{X} = (\mathscr{T}, N, S, \mathscr{T}, \langle \chi, \gamma \rangle)$ where χ is simply string concatenation, and the states S and local rules γ can be chosen freely.

Now every cell has the same successor structure: it has two children, each of which has two children, who both have two children in turn, and so on. So $\mathscr{T} \circledS x$ is isomorphic to $\mathscr{T} \circledS x'$ for any $x, x' \in \mathscr{T}$, which is also a coalgebra isomorphism $\overrightarrow{\mathscr{T} \circledS x} \to \overrightarrow{\mathscr{T} \circledS x'}$ since all cells have the same local rule. The only difference between x and x' is the number of ancestors they may have; but modal formulas can only refer to cells reachable via paths in $\mathscr{T}$, and there are no paths from x to its ancestors. Validity of modal formulas at x can only depend on cells reachable from x (this is proven in Lemma 8.7), i.e., only on $\mathscr{T} \circledS x$. Hence we conclude that *all* cells in $\mathscr{T}$ validate exactly the same formulas.

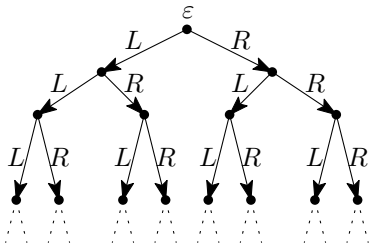


Figure 20: Visualisation of the freely generated binary tree monoid $\mathscr{T} = (\{L, R\}, \bullet, \varepsilon)$.

A similar observation holds for the forktree CA $\mathscr{F}$ (Figure 5), but now $x \leftrightsquigarrow x'$ iff $N_x = N_{x'}$. Hence there are two classes of logically equivalent cells on $\mathscr{F}$.

8.1.2 Motivation extending similarity with monoid homomorphisms

Example 8.2 (90°rotation)**.** Let $\mathbb{X} = (\mathbb{Z}, N, S, \mathcal{X})$ where $\mathcal{X} = (X, \langle \chi, \gamma \rangle)$ be a simple 1D CA frame, in particular with the 110 Wolfram number rule[1](the numbers were introduced by Wolfram in [50], but see [45,

§2.4] for this specific rule).

$$X = \{x_i\}_{i\in\mathbb{Z}} \qquad N = \{-1, 0, +1\}$$
$$n \circledS x_i = x_{i+n} \qquad S = \{\bullet, \circ\}$$
$$\gamma(\circ,\circ,\circ) = \circ \qquad \gamma(\circ,\circ,\bullet) = \bullet$$
$$\gamma(\circ,\bullet,\circ) = \bullet \qquad \gamma(\circ,\bullet,\bullet) = \bullet$$
$$\gamma(\bullet,\circ,\circ) = \circ \qquad \gamma(\bullet,\circ,\bullet) = \bullet$$
$$\gamma(\bullet,\bullet,\circ) = \bullet \qquad \gamma(\bullet,\bullet,\bullet) = \circ$$

Now make a copy $\mathcal{X}^j$ for every $j \in \mathbb{Z}$, and "stack up" these copies along the vertical axis of $\mathbb{Z}^2$ to obtain the CA $\mathcal{H} = (\{x_i^j\}_{i,j\in\mathbb{Z}}, \langle \chi^H, \{\gamma^j\}_{j\in\mathbb{Z}}\rangle)$ with $N^H = ((-1,0),(0,0),(1,0))$ and $(a,b) \circledS^H x_{i,j} = x_{i+a,j+b}$. This CA is simply a parallel execution of independent copies of $\mathcal{X}$.

Now we can repeat the same trick, but rotated by 90°counter clockwise (↺): let $\mathcal{V} = (\{x_i^j\}_{i,j\in\mathbb{Z}}, \langle \chi^V, \{\gamma^j\}_{j\in\mathbb{Z}}\rangle)$ with $N^V = ((0,-1),(0,0),(0,1))$ and $(a,b)\chi^V x_i^j = x_{i+b}^{j+a}$. We obtain the same parallel execution of the copies of $\mathcal{X}$, but with the whole graph rotated 90°.

Intuitively, $\mathcal{H}$ and $\mathcal{V}$ are "the same", but they are not similar in the sense defined above ($N^H \neq N^V$, so coalgebra homomorphisms $\mathcal{H} \to \mathcal{V}$ are not even defined). Nor do they satisfy the same logic, as evident from the fact that $\mathcal{H}$ validates

$$\psi = \big((\diamond_{(-1,0)}\bullet) \wedge (\diamond_{(0,0)}\circ) \wedge (\diamond_{(1,0)}\bullet)\big) \to \boxtimes\bullet, \tag{74}$$

i.e., the encoding of the rule "$\gamma(\bullet,\circ,\bullet) = \bullet$". This formula is *not* valid in $\mathcal{V}$, since the configuration $c_0 \colon V \to \{\bullet,\circ\}$ with around the origin is:

$c_0 =$

1	○	○	○
0	●	○	●
−1	○	○	○
	-	0	1

1

the next timestep will result in

$c_1 =$

1			
0	●	○	●
−1			
	-	0	1

1

(states that we cannot compute from the partial configuration are omitted).
Since c_0 satisfies the premise of ψ but not the consequence we conclude that $\mathbb{V}, c_0, x_0^0 \not\models \psi$.

The problem is that the neighbours of x_0^0 are above and below it, but left and right in $\mathcal{H}$ and in ψ. A simple fix is to apply a 90°counterclockwise rotation

$$\rho \colon \mathbb{Z}^2 \xrightarrow{\sim} \mathbb{Z}^2$$
$$\rho \colon (a,b) \mapsto (-b,a)$$

to all modalities. This gives

$$\rho[\psi] = \big((\diamond_{(0,-1)}\bullet) \wedge (\diamond_{(0,0)}\circ) \wedge (\diamond_{(0,1)}\bullet)\big) \to \boxtimes\bullet, \tag{75}$$

which *is* valid in $\mathcal{V}$.

[1] Here we slightly abuse the notation $\gamma(\circ,\bullet,\circ) = \bullet$ to denote $\gamma_{x_i}(z) = \bullet$ where $z(|-1|_x) = z(|1|_x) = \circ$ and $z(|0|_x) = \bullet$.

On the semantical side, ρ can also be extended to a variant of a coalgebra homomorphism $h\colon H \to V$, that rotates cell coordinates 90°: $h(x_i^j) \stackrel{\text{def}}{=} x_{-j}^i$. This mapping satisfies a property very similar to the $\mathbb{Z}^2$-$\circledS^H$-$\circledS^V$-homomorphism property (Definition 5.2) that coalgebra homomorphisms satisfy (as proven in Lemma 5.3):

$$h((a,b) \circledS^H x_i^j) = h(x_{i+1}^{j+b}) \stackrel{\text{def}}{=} x_{-j-b}^{i+a} = (-b,a) \circledS^V x_{-j}^i = \rho(a,b) \circledS^V h(x_i^j)$$

(For a real $\mathbb{Z}^2$-$\circledS^H$-$\circledS^V$-homomorphism $k\colon \mathcal{Y} \to \mathcal{Z}$, the property is of the form $k(m \circledS y) = m \circledS k(y)$, but here the RHS is of the form $\rho(m) \circledS k(y)$.) Observe that h preserves the dynamics of the CA; the $(-1,0)$ neighbour of a cell x_i^j, which is x_{i-1}^j is mapped to the $(0,-1)$ neighbour of x_{-j}^i, which is x_{-j}^{i-j}. Indeed, the left neighbour (horizontal layout of $\mathcal{X}$ in $\mathbb{Z}$) is mapped to the up neighbour (vertical layout), and the local rules are the same. Also, h has an inverse: rotating 90°clockwise.

Conclusion: we retain a similarity-to-logical-equivalence correspondence if (1) we generalise the definition of coalgebra homomorphisms to be paired with a monoid homomorphism, and (2) allow logical equivalence up to a monoid homomorphism on space modalities. Note that this is indeed a generalisation, since the old definition is recovered by taking the identity monoid homomorphism.

8.1.3 Motivation extending similarity with state mappings

A CA $\mathbb{Y}$ might be able to simulate a CA $\mathbb{X}$, while using a different state $s' \in S_Y$ to represent a certain simulated state $s \in S_X$.

As for a simple example, take the single-cell CA $\mathbb{X} = (\mathbb{1}, \mathbb{1}, S_X = \mathbb{N}, (X = \mathbb{1}, \langle \chi, \gamma \rangle))$ where the graph is trivial $\mathbf{1} \circledS \mathbf{1} = \mathbf{1}$ but the state becomes increment every timestep $\gamma_{\mathbf{1}}(\mathsf{const}_n) = n+1$. Let $\mathbb{Y}$ be the same CA, except for using $S_Y = \mathbb{N}_{\geq 1}$. Clearly, $\mathbb{X}$ becomes $\mathbb{Y}$ after incrementing all states by 1. This also holds for the logic, for example, $\mathbb{X}, \mathsf{const}_0, \mathbf{1} \vDash \boxtimes\boxtimes 2$ iff $\mathbb{Y}, \mathsf{const}_1, \mathbf{1} \vDash \boxtimes\boxtimes 3$. Note that const_1 is $i \circ \mathsf{const}_0$ where $i\colon n \mapsto n+1$, and that $\boxtimes\boxtimes 3$ is $\boxtimes\boxtimes 2$ with i applied to all occurrences of states.

A reverse simulation, of $\mathbb{Y}$ by $\mathbb{X}$, is obviously given by decrementing states by 1. But also incrementing states by any other fixed $n \in \mathbb{N}$ works; if $i_{1000}\colon n \mapsto n + 1000$, then $\mathbb{Y}, \mathsf{const}_1, \mathbf{1} \vDash \psi$ iff $\mathbb{X}, \mathsf{const}_{1001}, \mathbf{1} \vDash i_{1000}[\psi]$.

Conclusion: we obtain more bisimarity-to-logical-equivalences if we extend coalgebra homomorphisms (in this case the identity on cells) with a function between state sets $S_X \to S_Y$, and define logical equivalence also up to a map between states.

8.2 Basic Hennessy-Milner theorem

The logic $\mathcal{L}_{\mathsf{stg}}$ admits a cellular automaton variant of the Hennessey-Milner theorem of Kripke modal logics. The Hennessey-Milner theorem states that two worlds in two image-finite Kripke models satisfy the same formulas iff they are "bisimilar" [12]. We will define an alternative notion of similarity for CA cells, and prove two cells *validate* (i.e., satisfy on all models) the same formulas iff they are similar. The upcoming definitions assume that the compared CA have the same monoid; Section 8.3 will generalise the definitions to distinct monoids.

Definition 8.3: Similar cells

Let $\mathcal{X} = (X, \langle \chi, \gamma \rangle)$ and $\mathcal{Y} = (Y, \langle \xi, \delta \rangle)$ be CA in $\mathbf{CCA}_{\mathsf{G}}$ based on the same signature $(\mathscr{M}, N, S)$, and let $x \in X$ and $y \in Y$. Write $\overrightarrow{\mathscr{M} \circledS x}$ and $\overrightarrow{\mathscr{M} \circledS y}$ for the subcoalgebras of $\mathcal{X}$ and $\mathcal{Y}$ respectively containing the cells $\mathscr{M} \circledS x = \{m \circledS x \mid m \in \mathscr{M}\}$ and $\mathscr{M} \circledS y = \{m \circledS y \mid m \in \mathscr{M}\}$.

Then x *is similar to* y, denoted $x \leftrightarrows y$, iff there exist coalgebra homomorphisms

$$\overrightarrow{\mathscr{M} \circledS x} \underset{k}{\overset{h}{\rightleftarrows}} \overrightarrow{\mathscr{M} \circledS y}$$

such that $h(x) = y$ and $k(y) = x$.

Remark 8.4. This definition of similarity is very strict: observe that cells are already non-similar if they have a different number of neighbours (h and k are both surjective on the set of reachable cells, by Lemma 5.4) or if they have a different local rule.

Remark 8.5. The standard coalgebraic definition of bisimilarity states that cells x and y are bisimilar in $\mathcal{Z}$ if there exist coalgebra homomorphism $h\colon \mathcal{X} \to \mathcal{Z}$ and $k\colon \mathcal{Y} \to \mathcal{Z}$ such that $h(x) = k(y)$. Similarity according to Definition 8.3 hence means that x and y are both bisimilar in $\mathcal{X}$ and in $\mathcal{Y}$.

Definition 8.6: Logically equivalent

Let x be a cell in the frame $\mathbb{X} = (\mathscr{M}, N, \mathcal{X})$ and y be a cell in the frame $\mathbb{Y} = (\mathscr{M}, N, \mathcal{Y})$. Then x and y are logically equivalent, denoted as $x \leftrightsquigarrow y$, when for all $\psi \in \Psi$, $\mathbb{X}, x \vDash \psi$ iff $\mathbb{Y}, y \vDash \psi$.

Note that logical equivalence is about *validity* of a formula at a cell, not about *satisfiability*, since we are not using an initial configuration $c_0\colon X \to S$.

Whether a formula is satisfied in a cell x depends only on the cells that are reachable from a path from x. This is intuitively to be expected, since (1) the local rule of x depends only on cells that x can reach (since $N \subseteq \mathscr{M}$), and (2) we only have space modalities that follow existing paths in $\mathscr{M}$.[2]

Lemma 8.7

Let x be a cell in a frame $\mathbb{X} = (\mathscr{M}, N, X, \langle \chi, \gamma \rangle)$. Let $c, c'\colon X \to S$ s.t. If $c \restriction (\mathscr{M} \circledcirc x) = c' \restriction (\mathscr{M} \circledcirc x)$. Then for all $\psi \in \Psi$ it holds: $\mathbb{X}, c, x \vDash \psi$ iff $\mathbb{X}, c', x \vDash \psi$.

Proof. Prove the stronger claim: for all $t \in \mathbb{N}$, $m \in \mathscr{M}$ it holds that:

$$\mathbb{X}, c, m \circledcirc x \vDash \psi \text{ iff } \mathbb{X}, c', m \circledcirc x \vDash \psi$$

by induction on ψ.

□

Lemma 8.8

If there exist coalgebra homomorphisms

$$\overrightarrow{\mathscr{M} \circledcirc x} \underset{k}{\overset{h}{\rightleftarrows}} \overrightarrow{\mathscr{M} \circledcirc y}$$

such that $h(x) = y$ and $k(y) = x$, then h and k are inverses of each other.

Proof. For all $m \in \mathscr{M}$, it holds that:

$$\begin{aligned} k(h(m \circledcirc x)) &= k(m \circledcirc h(x)) \\ &= m \circledcirc k(h(x)) \\ &= m \circledcirc k(y) \\ &= m \circledcirc x. \end{aligned}$$

An analogous argument shows that $h(k(m \circledcirc y)) = m \circledcirc y$. □

[2]

Corollary 8.9

If there exist coalgebra homomorphisms

$$\overrightarrow{\mathscr{M} \otimes x} \underset{k}{\overset{h}{\rightleftarrows}} \overrightarrow{\mathscr{M} \otimes y}$$

such that $h(x) = y$ and $k(y) = x$, then for all $m \in \mathscr{M}$:

$$N_{m \otimes x} = N_{h(m \otimes x)}$$
$$N_{k(m \otimes y)} = N_{m \otimes y}$$

Proof. The statements are symmetric, so we only prove $N_{m \otimes x} = N_{h(m \otimes x)}$ explicitly.

Let $x' = m \otimes x$. We have $|n_1|_{x'} = |n_2|_{x'} \in N_{x'}$ iff $n_1 \otimes x = n_2 \otimes x$. Since h is a bijection (Lemma 8.8), this holds iff $n_1 \otimes h(x') = h(n_1 \otimes x') = h(n_2 \otimes x') = n_2 \otimes h(x')$, which holds iff $|n_1|_{h(x')} = |n_2|_{h(x')}$. □

The following lemma is (a slightly strengthened variant) of one direction of the Hennessey-Milner theorem (the other direction is proven in Theorem 8.12).

Lemma 8.10

If $x \leftrightarrows y$ then $x \leftrightsquigarrow y$.

Proof. We will prove the following two claims:

- For all $t \in \mathbb{N}$, $c_0 \colon X \to S$, $m \in \mathscr{M}$ and $\psi \in \Psi$:

$$\mathbb{X}, G^t_{\mathbb{X}}(c_0), m \otimes x \vDash \phi \text{ iff } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(m \otimes x) \vDash \phi \tag{H1}$$

- For all $t \in \mathbb{N}$, $c_0 \colon Y \to S$, $m \in \mathscr{M}$ and $\psi \in \Psi$:

$$\mathbb{Y}, G^t_{\mathbb{Y}}(c_0), m \otimes y \vDash \psi \text{ iff } \mathbb{X}, G^t_{\mathbb{X}}(c_0 \circ h), k(m \otimes y) \vDash \psi \tag{H2}$$

To see why these claims imply $x \leftrightsquigarrow y$, suppose $\mathbb{X}, x \vDash \psi$ and take any arbitrary $c_0 \colon Y \to S$.

Satisfiability of ψ at y only depends on $c_0 \restriction (\mathscr{M} \otimes y)$ (Lemma 8.7), we have $\mathbb{Y}, c_0, y \vDash \psi$ iff $\mathbb{Y}, c'_0 \circ k, y \vDash \psi$ for all $c'_0 \colon X \to S$ such that $c'_0 \circ k \restriction (\mathscr{M} \otimes y) = c_0 \restriction (\mathscr{M} \otimes y)$. From (H1) it follows that $\mathbb{Y}, c'_0 \circ k, y \vDash \psi$ holds iff $\mathbb{X}, c'_0, x \vDash \psi$, but that is known to hold (we assumed $\mathbb{X}, x \vDash \psi$). Since h and k are bijections (Lemma 8.8, we do have exactly such an c'_0, namely $c'_0 := c_0 \circ h$ (now $c'_0 \circ k = c_0$, so the requirement is satisfied trivially).

It remains to prove (H1) and (H2). Because of symmetry we only need to prove (H1) explicitly (the proof of (H2) is exactly the same after swapping the symbols $h \leftrightarrow k$, $X \leftrightarrow Y$, $x \leftrightarrow y$, $\chi \leftrightarrow \xi$ and $\gamma \leftrightarrow \delta$).

We proceed by induction on ψ. As a shorthand for induction hypotheses, write $H(\phi)$ to denote:

$$H(\phi) \overset{\text{def}}{=} \bigvee_{\substack{c_0 \colon X \to S \\ t \in \mathbb{N} \\ m \in \mathscr{M}}} \mathbb{X}, G^t_{\mathbb{X}}(c_0), m \otimes x \vDash \phi \text{ iff } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(m \otimes x) \vDash \phi \tag{76}$$

Case $\psi = \phi_1 \vee \phi_2$
Induction hypotheses: $H(\phi_1)$ *and* $H(\phi_2)$.

We observe that, for all t, c_0 and $x' = m \circledast x \in (\mathscr{M} \circledast x)$, it holds that:

$$\begin{aligned}
&\mathbb{X}, G^t_{\mathbb{X}}(c_0), x' \vDash \phi_1 \vee \phi_2 \\
&\qquad \text{iff } (\mathbb{X}, G^t_{\mathbb{X}}(c_0), x' \vDash \phi_1 \text{ or } \mathbb{X}, G^t_{\mathbb{X}}(c_0), x' \vDash \phi_2) && \text{// Semantics of } \vee. \\
&\qquad \text{iff } (\mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(x') \vDash \phi_1 \text{ or } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(x') \vDash \phi_2) && \text{// Induction hypothesis.} \\
&\qquad \text{iff } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(x') \vDash \phi_1 \vee \phi_2 && \text{// Semantics of } \vee.
\end{aligned}$$

Case $\psi = \bigvee_{i \in I} \phi_i$
Induction hypotheses: $H(\phi_i)$ *for all* $i \in I$.
As in the binary disjunction case, for all t, c_0 and $x' = m \circledast x$ it holds that:

$$\begin{aligned}
&\mathbb{X}, G^t_{\mathbb{X}}(c_0), x' \vDash \bigvee_{i \in I} \psi_i \\
&\qquad \text{iff } \text{ exists } i \in I \text{ such that } \mathbb{X}, G^t_{\mathbb{X}}(c_0), x' \vDash \phi_i && \text{// Semantics of infinitary disjunctions.} \\
&\qquad \text{iff } \text{ exists } i \in I \text{ such that } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(x') \vDash \phi_i && \text{// Induction hypothesis.} \\
&\qquad \text{iff } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(x') \vDash \bigvee_{i \in I} \psi_i. && \text{// Semantics of } \vee.
\end{aligned}$$

Case $\psi = \bot$
It follows immediately from the semantics of $\bot$ that, for all t, c_0 and $x' = m \circledast x$, that both $\mathbb{X}, G^t_{\mathbb{X}}(c_0), x' \vDash \bot$ and $\mathbb{Y}, G^t_{\mathbb{Y}}(c_o \circ k), h(x') \vDash \bot$ are always false.

Case $\psi = \neg\phi$
Induction hypothesis: $H(\phi)$.
For all t, c_0 and $x' = m \circledast x$ it holds that:

$$\begin{aligned}
\mathbb{X}, G^t_{\mathbb{X}}(c_0), x' \vDash \neg\phi \text{ iff } &\mathbb{X}, G^t_{\mathbb{X}}(c_0), x' \nvDash \phi \\
\text{iff } &\mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(x') \nvDash \phi && \text{// Induction hypothesis.} \\
\text{iff } &\mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(x') \vDash \neg\phi
\end{aligned}$$

Case $\psi = \mathsf{v}_s$ **for some** $s \in S$
This case is still quite straightforward, since the semantics of v_s do not depend on t. In particular, for all t, c_0 and $x' = m \circledast x$, it holds that:

$$\begin{aligned}
\mathbb{X}, G^t_{\mathbb{X}}(c_0), x' \vDash \mathsf{v}_s \text{ iff } &c_0(x') = s \\
\text{iff } &c_0(k(h(x'))) = s && \text{// } k(h(x')) = x' \text{ by Lemma 8.8.} \\
\text{iff } &\mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(x') \vDash \mathsf{v}_s
\end{aligned}$$

Case $\psi = s$ **for some** $s \in S$
This case is considerably more complicated than the v_s case, since the truth value of s *does* depend on the point in time. Inductively we need to show that $c_t(x') = (c_t \circ k)(h(x'))$ (where $c_t = G^t_{\mathbb{X}}(c_0)$) for all $x' \in \mathscr{M} \circledast x$ and $t \in \mathbb{N}$, which is done in Lemma 8.11 below. With this lemma available, we can show (for all t, c_0 and $x' = m \circledast x$) that:

$$\begin{aligned}
&\mathbb{X}, G^t_{\mathbb{X}}(c_0), x' \vDash s \\
&\qquad \text{iff } G^t_{\mathbb{X}}(c_0)(x') = s && \text{// Semantics of } s. \\
&\qquad \text{iff } G^t_{\mathbb{Y}}(c_0 \circ k)(h(x')) = s && \text{// Lemma 8.11.} \\
&\qquad \text{iff } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(x') \vDash s. && \text{// Semantics of } s.
\end{aligned}$$

Case $\psi = \Diamond_n \phi$ for some $n \in \mathscr{M}$
Induction hypothesis: $H(\phi)$.
For all t, c_0 and m it holds that:

$$\begin{aligned}
&\mathbb{X}, G^t_{\mathbb{X}}(c_0), m \circledS x \vDash \Diamond_n \phi \\
&\qquad \text{iff } \mathbb{X}, G^t_{\mathbb{X}}(c_0), n \circledS m \circledS x \vDash \phi && \text{// Semantics } \Diamond_n. \\
&\qquad \text{iff } \mathbb{X}, G^t_{\mathbb{X}}(c_0), (n \bullet m) \circledS x \vDash \phi \\
&\qquad \text{iff } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h((n \bullet m) \circledS x) \vDash \phi \\
&\qquad \text{// Induction hypothesis (note that } (n \bullet m) \in \mathscr{M}, \text{ so } (n \bullet m) \circledS x \in (\mathscr{M} \circledS x)). \\
&\qquad \text{iff } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(n \circledS m \circledS x) \vDash \phi \\
&\qquad \text{iff } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), n \circledS h(m \circledS x) \vDash \phi \\
&\qquad \text{iff } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(m \circledS x) \vDash \Diamond_n \phi. && \text{// Semantics } \Diamond_n.
\end{aligned}$$

Case $\psi = \boxtimes \phi$
Induction hypothesis: $H(\phi)$.
This is the case for which the quantification over time in (H1) was needed; for all t, c_0 and $x' = m \circledS x$ it now holds that:

$$\begin{aligned}
&\mathbb{X}, G^t_{\mathbb{X}}(c_0), x' \vDash \boxtimes \phi \\
&\qquad \text{iff } \mathbb{X}, G^{t+1}_{\mathbb{X}}(c_0), x' \vDash \phi && \text{// Semantics } \boxtimes. \\
&\qquad \text{iff } \mathbb{Y}, G^{t+1}_{\mathbb{Y}}(c_0 \circ k), h(x') \vDash \phi && \text{// Induction hypothesis.} \\
&\qquad \text{iff } \mathbb{Y}, G^t_{\mathbb{Y}}(c_0 \circ k), h(x') \vDash \boxtimes \phi. && \text{// Semantics } \boxtimes.
\end{aligned}$$

□

Lemma 8.11

If there exist coalgebra homomorphisms

$$\overrightarrow{\mathscr{M} \circledS x} \underset{k}{\overset{h}{\rightleftarrows}} \overrightarrow{\mathscr{M} \circledS y}$$

such that $h(x) = y$ and $k(y) = x$, then it holds for all $t \in \mathbb{N}$ and $c_0 \colon X \to S$ that:

$$G^t_{\mathbb{X}}(c_0) \restriction (\mathscr{M} \circledS x) = \big(G^t_{\mathbb{Y}}(c_0 \circ k) \circ h\big) \restriction (\mathscr{M} \circledS x). \tag{77}$$

Proof. Induction on $t \in \mathbb{N}$. The base case, when $t = 0$, follows directly from Lemma 8.8 and the fact that $G^0 = \mathsf{Id}$:

$$c_0(m \circledS x) = c_0(k(h(m \circledS x)) = ((c_0 \circ k) \circ h)(m \circledS x)$$

holds for all $m \circledS x \in \mathscr{M} \circledS x$.
For the inductive case, assume (77) holds for a given $t \in \mathbb{N}$. We need to show it also holds for $t+1$. Take an

arbitrary $x' = m \otimes x \in \mathscr{M} \otimes x$, then we compute:

$$\begin{aligned}
G^{t+1}_{\mathbb{X}}(c_0)(x') &= G_{\mathbb{X}}(G^t_{\mathbb{X}}(c_0))(x') \\
&= \gamma_{x'}(\lambda\, |n|_{x'} \;.\; G^t_{\mathbb{X}}(c_0)(n \otimes x')) && \text{// Definition of global rule (Eq. (28)).} \\
&= \delta_{h(x')}(\lambda\, |n|_{h(x')} \;.\; G^t_{\mathbb{X}}(c_0)(h(n \otimes x'))) && \text{// By Corollary 8.9, } |n|_{x'} = |n|_{h(x')}. \\
& && \text{// Thus by the extension property of coalgebra homomorphisms (Theorem 5.9).} \\
&= \delta_{h(x')}(\lambda\, |n|_{h(x')} \;.\; G^t_{\mathbb{X}}(c_0)(n \otimes h(x'))) \\
&= \delta_{h(x')}(\lambda\, |n|_{h(x')} \;.\; G^t_{\mathbb{Y}}(c_0 \circ k)(n \otimes h(x'))) && \text{// Induction hypothesis.} \\
&= G_{\mathbb{Y}}(G^t_{\mathbb{Y}}(c_0 \circ k)(h(x')) && \text{// Definition of global rule (Eq. (28)).} \\
&= G^{t+1}_{\mathbb{Y}}(c_0 \circ k)(h(x')).
\end{aligned}$$

□

The converse of Lemma 8.8 also holds, but in $\mathcal{L}_{\mathsf{stg}}$ only if neighbourhoods N and M are finite:

Theorem 8.12

Let x be a cell in the frame $\mathbb{X} = (\mathscr{M}, N, \mathcal{X})$ and y be a cell in the frame $\mathbb{Y} = (\mathscr{M}, N, \mathcal{Y})$. then $x \leftrightarrows y$ iff $x \leftrightsquigarrow y$. (This holds in $\mathcal{L}'_{\mathsf{CA}}$ and $\mathcal{L}_{\mathsf{CA}}$, and also in $\mathcal{L}_{\mathsf{stg}}$ if N and M are finite).

Proof. The left-to-right direction follows from Lemma 8.8. For the other direction, assume $x \leftrightsquigarrow y$. We now need to construct coalgebra homomorphisms

$$\overrightarrow{\mathscr{M} \otimes x} \underset{k}{\overset{h}{\rightleftarrows}} \overrightarrow{\mathscr{M} \mathbin{⅋} y}$$

such that $h(x) = y$ and $k(y) = x$. By Theorem 5.9, these need to be $\mathscr{M}$-χ-ξ-homomorphisms. Hence we have no choice but to define (for all $m \in \mathscr{M}$):

$$\begin{aligned}
&h\colon x \mapsto y, \\
&h\colon m \otimes x \mapsto m \mathbin{⅋} h(x) = m \mathbin{⅋} y, \\
&k\colon y \mapsto x, \\
&k\colon m \mathbin{⅋} y \mapsto m \otimes k(y) = m \otimes x.
\end{aligned}$$

Note that $h = k^{-1}$.

It remains to prove the extension property. Because of symmetry, we only show this for h; so it remains to prove that for all $x' = m \otimes x \in \mathscr{M} \otimes x$ it holds that:

$$\gamma_{x'}(z \lVert x') = \delta_{h(x')}(z \lVert h(x'))$$

for all $z\colon N \to S$ that are $N_{h(x')}$ compatible. Note that $N_{x'} = N_{h(x')}$ since h is an isomorphism (isomorphism implies $n_1 \otimes x' = n_2 \otimes x'$ iff $n_1 \mathbin{⅋} h(x') = h(n_1 \otimes x') = h(n_2 \otimes x') = n_2 \mathbin{⅋} h(x')$, i.e., $|n_1|_{x'} = |n_2|_{x'}$ iff $|n_1|_{h(x')} = |n_2|_{h(x')}$).

Further observe that

$$\mathsf{Log}(m \otimes x) = \mathsf{Log}(h(m \otimes x)). \tag{78}$$

This follows from the fact that (omitting frames) $m \otimes x \vDash \phi$ iff $x \vDash \diamond_m \phi$ iff (because $x \leftrightsquigarrow y$) $y \vDash \diamond_m \phi$ iff $m \mathbin{⅋} y \vDash \phi$ and $m \mathbin{⅋} y = m \mathbin{⅋} h(x) = h(m \otimes x)$ (these statements can easily be proven via the semantics of $\diamond_m$).

We can use Lemma 6.12 to express $\gamma_{x'}(z \lVert x')$ as a formula (in $\mathcal{L}'_{\mathsf{CA}}$ and $\mathcal{L}_{\mathsf{CA}}$ this is possible for all N, in $\mathcal{L}_{\mathsf{stg}}$ only if N is finite). In particular, $\gamma_{x'}(z \lVert x') = s_0$ holds iff

$$\mathbb{X}, x' \vDash \left(\bigwedge_{n \in N} \Diamond_n z(|n|_{x'}) \right) \to \mathbb{Z} s_0,$$

which holds iff (by Eq. (78), and since $|n|_{h(x')} = |n|_{x'}$)

$$\mathbb{Y}, h(x') \vDash \left(\bigwedge_{n \in N} \Diamond_n z(|n|_{h(x')}) \right) \to \mathbb{Z} s_0,$$

which (again by Lemma 6.12) holds iff $\delta_{h(x')}(z \lVert h(x')) = s_0$. Since z and s_0 were arbitrary, the desired conclusion also holds: $\gamma_{x'}(z \lVert x') = \delta_{h(x')}(z \lVert h(x'))$ for all $N_{x'} = N_{h(x')}$ compatible configuration $z \colon N \to S$. □

8.3 Generalised Hennessy-Milner theorem

Theorem 8.12 gives a variant of the Hennessy-Milner theorem that allows to compare similarity of two CA $\mathcal{X}$ and $\mathcal{Y}$ that are based on the same monoid $\mathscr{M}$ and use the same set of states S. We will now formalise the generalised definitions of similarity and logical equivalence as sketched in Sections 8.1.2 and 8.1.3, and prove a corresponding Hennessey-Milner theorem.

8.3.1 General CA morphisms

We will now generalise the definition of coalgebra homomorphisms to include morphisms between CA defined on different signatures. Recall that $\mathbf{CCA}_{\mathsf{G}}$ (Definition 4.16) is the coproduct of $\mathbf{CCA}^{\Sigma}_{\mathsf{G}}$ over all signatures Σ. As such, it has no morphisms between CA defined on different signatures.

We will extend $\mathbf{CCA}_{\mathsf{G}}$ to the category $\mathbf{CCA}^{+}_{\mathsf{G}}$, which has the same objects, but whose morphisms are *general CA morphisms*, which have the characteristic properties of coalgebra homomorphisms (Theorem 5.9) but also include a monoid homomorphisms and a function between state sets. Coalgebra homomorphisms are special cases of general CA morphisms (where the monoid homomorphism and state set maps are the identities), so $\mathbf{CCA}_{\mathsf{G}}$ is a subcategory of $\mathbf{CCA}^{+}_{\mathsf{G}}$.

To define general CA morphisms we first need the following auxiliary definition:

Definition 8.13: Graph-neighbourhood morphism

Let $\mathscr{M}, \mathscr{N} \in \mathbf{Mon}$ be monoids and let $M \subseteq \mathscr{M}$ and $N \subseteq \mathscr{N}$. A *graph-neighbourhood morphism* $h_{\mathsf{G}} \colon (\mathscr{M}, M) \to (\mathscr{N}, N)$ is a monoid homomorphism (in $\mathbf{Mon}$) such that $h_{\mathsf{G}}[M] \subseteq N$.

Definition 8.14: General CA morphism

A *general CA morphism* $h \colon \mathbb{X} \to \mathbb{Y}$ from $\mathbb{X} = (\mathscr{M}, M, S_X, X, \langle \chi, \gamma \rangle)$ to $\mathbb{Y} = (\mathscr{N}, N, S_Y, Y, \langle \xi, \delta \rangle)$ is triple of functions $(h_{\mathsf{G}}, h_{\mathsf{C}}, h_{\mathsf{S}})$ where

- $h_{\mathsf{G}} \colon (\mathscr{M}, M) \to (\mathscr{N}, N)$ is a graph-neighbourhood morphism.
- $h_{\mathsf{C}} \colon X \to Y$ is a function between cells. We usually omit the subscript and write $h(x)$ for $h_{\mathsf{C}}(x)$.
- $h_{\mathsf{S}} \colon S_X \to S_Y$ is a function between states.

Furthermore, this triple adheres to the following constraints:

1. (**Module property**)
$$h_{\mathsf{C}}(m \circledast x) = h_{\mathsf{G}}(m) \circledcirc h_{\mathsf{C}}(x)$$
for all $x \in X$ and $m \in \mathscr{M}$.

2. h_{S} is injective.

3. For any $x \in X$ and $y \in Y$ s.t. $h_{\mathsf{C}}(x) = y$, the restriction $h_{\mathsf{C}} \restriction (\mathscr{M} \circledast x) \colon (\mathscr{M} \circledast x) \rightarrowtail (\mathscr{N} \circledcirc y)$ is an injection.

4. (**Extension property**)
$$h_{\mathsf{S}}(\gamma_x(z)) = \delta_{h_{\mathsf{C}}(x)}(z')$$
for every $x \in X$ and every $z \colon M_x \to S_X$, where $z' \colon N_{h(x)} \to S_Y$ is any extension of $h_{\mathsf{S}} \circ z \circ (h_{\mathsf{G}}^{M_x})^{-1}$. Here $h_{\mathsf{G}}^{M_x} \colon M_x \to N_{h_{\mathsf{C}}(x)}$ is defined as $h_{\mathsf{G}}^{M_x} \colon |m|_x \mapsto |h_{\mathsf{G}}(m)|_{h_{\mathsf{C}}(x)}$[3].
As a commuting diagram this is:

$$\begin{array}{ccccc} S_X^{M_x} & \xrightarrow{h_{\mathsf{S}} \circ (-) \circ (h_{\mathsf{G}}^{M_x})^{-1}} & S_Y^{h_{\mathsf{G}}^{M_x}[M_x]} & \xrightarrow{\text{extend}} & S_Y^{N_{h(x)}} \\ {\scriptstyle\gamma}\downarrow & & & & \downarrow{\scriptstyle\delta_{h(x)}} \\ S_X & & \xrightarrow[h_{\mathsf{S}}]{} & & S_Y \end{array}$$

A coarse shorthand to capture the essence of this law is:
$$h_{\mathsf{S}}\gamma = \delta h_{\mathsf{S}}.$$

Note that coalgebra homomorphisms are special cases of general CA homomorphisms.

8.3.2 Generalised notions of similarity

Definition 8.15: Simulation and mutual similarity

Let $\mathbb{X} = (\mathscr{M}, M, S_X, X, \langle \chi, \gamma \rangle)$ and $\mathbb{Y} = (\mathscr{N}, N, S_Y, Y, \langle \xi, \delta \rangle)$ be frames in $\mathbf{CCA}_{\mathsf{G}}^{+}$. A cell $x \in X$ *is simulated by* a cell $y \in Y$, denoted as $x \rightrightarrows y$, if there exists a general CA morphism $h \colon \overrightarrow{\mathscr{M} \circledast x} \to \overrightarrow{\mathscr{M} \circledcirc y}$ such that $h_{\mathsf{C}}(x) = y$. If also $y \rightrightarrows x$ (i.e., if there also exists a $k \colon \overrightarrow{\mathscr{M} \circledcirc y} \to \overrightarrow{\mathscr{M} \circledast x}$ such that $k_{\mathsf{C}}(y) = x$), then x *is mutually similar to* y, denoted as $x \leftrightarrows y$.

Remark 8.16. The definition of a simulation $h \colon x \rightrightarrows y$ tells directly that the local rules γ and δ commute with the embedding given by h. We will later see (in Lemma 8.24) that h also commutes with the global rules $G_{\mathbb{X}}$ and $G_{\mathbb{Y}}$.

Remark 8.17. The definition of mutual similarity does not require h and k to be each other's inverse.

The general definition of logical equivalence is less straightforward than Definition 8.3, since the two CA may have different states and different path monoids, and hence different symbols. We propose the following "translation" of formulas between incompatible CA:

Definition 8.18: Logical simulation

Let $\Psi(\mathscr{M}, M, S)$ denote the $\mathcal{L}_{\mathsf{stg}}$-formulas (idem when using $\mathcal{L}'_{\mathsf{CA}}$ or $\mathcal{L}_{\mathsf{CA}}$) defined on the monoid $\mathscr{M}$,

[3] It may not be obvious that $h_{\mathsf{G}}^{M_x}$ is well-defined, since $|h_{\mathsf{G}}(m)|_{h(x)}$ is ill-defined when $h_{\mathsf{G}}(m) \notin N$. The latter cannot occur, since the $h_{\mathsf{G}}[M] \subseteq N$ condition of a graph-neighbourhood morphism (Definition 8.13) ensures that $h_{\mathsf{G}}(m) \in N$ when $m \in M$.

neighbourhood $M \subseteq \mathscr{M}$ and state set S. Then a *logical simulation* $h_{\mathsf{L}} \colon \Psi(\mathscr{M}, M, S) \to \Psi(\mathscr{N}, N, S')$ is a pair $(h_{\mathsf{G}}, h_{\mathsf{S}})$ where $h_{\mathsf{G}} \colon (\mathscr{M}, M) \to (\mathscr{N}, N)$ is a graph-neighbourhood morphism (Definition 8.13) and $h_{\mathsf{S}} \colon S \to S'$ a function.

The computation of h_{L} on a formula in $\Psi(\mathscr{M}, S)$ is defined by straightforward recursion:

$$\begin{aligned}
h_{\mathsf{L}}(\bot) &\stackrel{\text{def}}{=} \bot \\
h_{\mathsf{L}}(s_i) &\stackrel{\text{def}}{=} h_{\mathsf{S}}(s_i) \\
h_{\mathsf{L}}(\mathsf{v}_{s_i}) &\stackrel{\text{def}}{=} \mathsf{v}_{h_{\mathsf{S}}(s_i)} \\
h_{\mathsf{L}}(\neg\psi) &\stackrel{\text{def}}{=} \neg h_{\mathsf{L}}(\psi) \\
h_{\mathsf{L}}(\psi_1 \vee \psi_2) &\stackrel{\text{def}}{=} h_{\mathsf{L}}(\psi_1) \vee h_{\mathsf{L}}(\psi_2) \\
h_{\mathsf{L}}(\bigvee i \in I \psi_i) &\stackrel{\text{def}}{=} \bigvee_{i \in I} h_{\mathsf{L}}(\psi_i) \\
h_{\mathsf{L}}(\diamond_m \psi) &\stackrel{\text{def}}{=} \diamond_{h_{\mathsf{G}}(m)} h_{\mathsf{L}}(\psi) \\
h_{\mathsf{L}}(\between \psi) &\stackrel{\text{def}}{=} \between h_{\mathsf{L}}(\psi)
\end{aligned}$$

We define the *image configuration* of h_{L} on $\mathbb{Y}$ as the set

$$\mathsf{Im}\mathsf{C}(h_{\mathsf{L}}) \stackrel{\text{def}}{=} \{c_0 \colon Y \to S_Y \mid c_0(h_{\mathsf{G}}(x)) = h_{\mathsf{S}}(s) \text{ for some } s \in S_X, \text{ for all } x \in X\}. \tag{79}$$

This are, intuitively, all possible extensions of the $(h_{\mathsf{G}}, h_{\mathsf{S}})$ images of $\mathbb{X}$ configurations.

To define logical equivalence, we first define logical simulation. Intuitively, $x \rightsquigarrow y$ if $\mathsf{Log}(x)$ can injectively be translated into a fragment of $\mathsf{Log}(y)$. It is possible that $\mathbb{Y}$ has more states or more paths than $\mathbb{X}$ does, but these are then simply not part of the fragment in which $\mathsf{Log}(x)$ is embedded. We call x and y if their logics can be embedded into one each other's. The formal definitions are as follows:

Definition 8.19: Logical simulation

Let $\mathbb{X} = (\mathscr{M}, M, S_X, X, \langle \chi, \gamma \rangle)$ and $\mathbb{Y} = (\mathscr{N}, N, S_Y, Y, \langle \xi, \delta \rangle)$ be frames in $\mathbf{CCA}^+_{\mathsf{G}}$. A cell $x \in X$ *is logically simulated by* a cell $y \in Y$, denoted as $x \rightsquigarrow y$, if there exists a logical simulation $h_{\mathsf{L}} \colon \Psi(\mathscr{M}, M, S_X) \to \Psi(\mathscr{N}, N, S_Y)$ such that for all $c_0 \colon Y \to S_Y \in \mathsf{Im}\mathsf{C}(h_{\mathsf{L}})$:

$$\mathbb{X}, h_{\mathsf{S}}^{-1} \circ c_0 \circ h_{\mathsf{C}}, x \vDash \psi \text{ iff } \mathbb{Y}, c_0, y \vDash h_{\mathsf{L}}(\psi) \tag{80}$$

for all $\psi \in \Psi(\mathscr{M}, S_X)$, where

$$\begin{aligned}
&h_{\mathsf{C}} \colon (\mathscr{M} \otimes x) \xrightarrow{\sim} (h_{\mathsf{G}}[\mathscr{M}] \circledS\, y), \\
&h_{\mathsf{C}} \colon m \otimes x \mapsto h_{\mathsf{G}}(m) \circledS\, y.
\end{aligned}$$

If both $x \rightsquigarrow y$ and $y \rightsquigarrow x$, then x and y are *mutually logically similar*, which is denoted by $x \leftrightsquigarrow y$.

Remark 8.20. Eq. (80) does apply to all models on the frame $\mathbb{X}$: Lemma 8.22 will show that h_{C} is injective, so every $c_0' \colon X \to S_X$ can be written in the form $c_0' = h_{\mathsf{S}}^{-1} \circ c_0 \circ h_{\mathsf{C}}$ for some $c_0 \in \mathsf{Im}\mathsf{C}(h_{\mathsf{L}})$. Note that this is well defined (despite h_{S} not being a bijection), as by definition of $\mathsf{Im}\mathsf{C}(h_{\mathsf{L}})$, all cells in the image of h_{C} do have a state in the image of h_{S}.

Time invariance (Definition 6.14) is preserved under logical simulations, for configurations using only states in $h_{\mathsf{S}}[S_X]$:

Lemma 8.21

If $h_{\mathsf{L}}\colon x \rightsquigarrow y$ and if ψ is time invariant for x then $h_{\mathsf{L}}(\psi)$ is time invariant for y on models using configurations $c_0\colon Y \to h_{\mathsf{S}}[S_X]$.

Proof. Let $c_0\colon Y \to h_{\mathsf{S}}[S_X]$, then (since h_{S} is injective) it is of the form $h_{\mathsf{S}} \circ c_0'$ for some $c_0'\colon Y \to S_X$. Hence $\mathbb{Y}, c_0, y \vDash h_{\mathsf{L}}(\psi)$ iff $\mathbb{X}, c_0' \circ h_{\mathsf{C}}, x \vDash \psi$. Since ψ is time invariant for x, the latter holds iff $\mathbb{X}, c_0' \circ h_{\mathsf{C}}, x \vDash \boxtimes^t \psi$ for all $t \in \mathbb{N}$. Since $x \rightsquigarrow y$, this in turn holds iff $\mathbb{Y}, c_0, x \vDash \boxtimes^t h_{\mathsf{L}}(\psi)$ for all $t \in \mathbb{N}$ (note that $h_{\mathsf{L}}(\boxtimes^t \psi) = \boxtimes^t h_{\mathsf{L}}(\psi)$). Hence $\mathbb{Y}, c_0, y \vDash h_{\mathsf{L}}(\psi)$ iff $\mathbb{Y}, c_0, x \vDash \boxtimes^t h_{\mathsf{L}}(\psi)$ for all $t \in \mathbb{N}$, which is exactly the property of time invariance. □

8.3.3 Generalised Hennessy-Milner theorem

We will now show that $x \leftrightarrows y$ iff $x \leftrightsquigarrow y$. First we prove the left-to-right direction. Although I state it as a lemma, in reality I first performed the proof, and used the result to find the correct definition for $x \rightrightarrows y$:

Lemma 8.22

If $x \rightsquigarrow y$ then $x \rightrightarrows y$, under the conditions that that $\mathsf{card}(S_X) \geq 2$ and $\mathsf{card}(S_Y) \geq 2$ (and in case of $\mathcal{L}_{\mathsf{stg}}$, also that M and N are finite).

Proof. The assumption $x \rightsquigarrow y$ gives us a logical simulation $h_{\mathsf{L}} = (h_{\mathsf{G}}, h_{\mathsf{S}})\colon \Psi(\mathscr{M}, M, S_X) \to \Psi(\mathscr{N}, N, S_Y)$. We will extend $(h_{\mathsf{G}}, h_{\mathsf{S}})$ to a general CA morphism $h = (h_{\mathsf{G}}, h_{\mathsf{C}}, h_{\mathsf{S}})\colon \overrightarrow{\mathscr{M} \circledS x} \to \overrightarrow{\mathscr{M} \circledR y}$. The definition of $h_{\mathsf{C}}\colon (\mathscr{M} \circledS x) \to (\mathscr{N} \circledR y)$ is straightforward:

$$h_{\mathsf{C}}\colon x \mapsto y$$
$$h_{\mathsf{C}}\colon (m \circledS x) \mapsto h_{\mathsf{G}}(m) \circledR y$$

This definition already implies that property 1 of a general CA morphism (Definition 8.14) is satisfied.

As for the injectivity of h_{S} (property 2), suppose for contradiction that there exist two states $s_0 \neq s_1 \in S_X$ such that $h_{\mathsf{S}}(s_0) = h_{\mathsf{S}}(s_1)$. In any configuration, x can have at most one state, so $\mathbb{X}, x \vDash s_0 \to \neg s_1$ (see Lemma 6.24). From $x \rightsquigarrow y$ it follows that $\mathbb{Y}, c_0, y \vDash h_{\mathsf{S}}(s_0) \to \neg h_{\mathsf{S}}(s_1)$ (for any $c_0\colon \mathsf{ImC}(h_{\mathsf{L}})$, see Remark 8.20). Now write $s' = h_{\mathsf{S}}(s_0)$, then this is $\mathbb{Y}, c_0, y \vDash s' \to \neg s'$. Now pick the particular $c_0' \in \mathsf{ImC}(h_{\mathsf{L}})$ such that $c_0'(y) = s'$ (e.g., $c_0' = \mathsf{const}_{s'}$). Then $\mathbb{Y}, c_0', y \vDash s'$, and from modus ponens (Lemma 6.21) it follows what $\mathbb{Y}, c_0', y \vDash \neg s'$. Thus $s' = c_0'(y) \neq s'$; a contradiction.

We show injectivity of h_{C} (property 3) via a similar argument. First observe that, for any $x' = m \circledS x \in \mathscr{M} \circledS x$:

$$\text{if } \mathbb{X}, x' \vDash \phi \text{ then } \mathbb{X}, x \vDash \Diamond_m \phi. \tag{81}$$

. Thus $\mathbb{Y}, c_0, y \vDash \Diamond_{h_{\mathsf{G}}(m)} h_{\mathsf{L}}(\phi)$ (for all $c_0 \in \mathsf{ImC}(h_{\mathsf{L}})$), which is equivalent to $\mathbb{Y}, c_0, h_{\mathsf{G}}(m) \circledR y \vDash h_{\mathsf{L}}(\phi)$. But by definition of h_{C}, it holds that $h_{\mathsf{C}}(x') = h_{\mathsf{G}}(m) \circledR y$, so $\mathbb{Y}, c_0, h_{\mathsf{C}}(x') \vDash h_{\mathsf{L}}(\phi)$.

Now suppose h_{C} is not injective, and that $h_{\mathsf{C}}(m_1 \circledS x) = h_{\mathsf{C}}(m_2 \circledS x)$ while $m_1 \circledS x \neq m_2 \circledS x$. Then $h_{\mathsf{G}}(m_1) \circledR y$ and $h_{\mathsf{G}}(m_2) \circledR y$ are the same cell, so they have always the same state. Hence $\mathbb{Y}, c_0, y \vDash (\Diamond_{h_{\mathsf{G}}(m_1)} h_{\mathsf{S}}(s)) \leftrightarrow (\Diamond_{h_{\mathsf{G}}(m_2)} h_{\mathsf{S}}(s))$ for any $s \in S_X$ and all $c_0 \in \mathsf{ImC}(h_{\mathsf{L}})$. By definition of $x \rightsquigarrow y$, this means that for all $s \in S$:

$$\mathbb{X}, x \vDash (\Diamond_{m_1} s) \leftrightarrow (\Diamond_{m_2} s) \tag{82}$$

(by Remark 8.20 we indeed obtain satisfaction in all models, i.e., validity). Since S_X has at least two states, so there exists $s_1 \neq s_2 \in S_X$. Recall that $m_1 \circledS x \neq m_2 \circledS x$ are different cells, we can pick $c_0'\colon X \to S_X$ such that $c_0'(m_1 \circledS x) = s_1$ and $c_0'(m_2 \circledS x) = s_2$. Then $\mathbb{X}, c_0', x \vDash \Diamond_{m_1} s_1$ by the semantics $\Diamond_{m_1} s_1$. But $\wedge$-elimination

and MP[4]we obtain $\mathbb{X}, c'_0, x \vDash \diamond_{m_2} s_1$, which holds iff $\mathbb{X}, c'_0, m_2 \circledast x \vDash s_1$, i.e., iff $c'_0(m_2 \circledast x) = s_1$; the latter is indeed a contradiction!

It remains to prove the extension property (property 4). To this end, take any arbitrary $x' = m \circledast x \in \mathscr{M} \circledast x$ and any $z \colon M_{x'} \to S_X$.
Let $z' \colon N_{h_{\mathsf{C}}(x')} \to S_Y$ be any arbitrary extension of $h_{\mathsf{S}} \circ z \circ (h_{\mathsf{G}}^{M_{x'}})^{-1} \colon h_{\mathsf{G}}^{M_{x'}}[M_{x'}] \to S_Y$. Using Lemma 6.12, we can express the local rule $\gamma_{x'}$ on input z as the formula:

$$\mathbb{X}, x' \vDash \left(\bigwedge_{|m_i| \in M_{x'}} \diamond_{m_i} z(|m_i|) \right) \to \boxtimes \gamma_{x'}(z). \tag{83}$$

We can do the same for $h_{\mathsf{C}}(x')$ and $\delta_{h_{\mathsf{C}}(x')}$:

$$\mathbb{Y}, h_{\mathsf{C}}(x') \vDash \left(\bigwedge_{|n| \in N_{h_{\mathsf{C}}(x')}} \diamond_n z'(|n|) \right) \to \boxtimes \delta_{h_{\mathsf{C}}(x')}(z'). \tag{84}$$

But, similarly as we did for Eq. (81), we can translate Eq. (83) under the logical simulation to:

$$\mathbb{Y}, c_0, h_{\mathsf{C}}(x') \vDash \left(\bigwedge_{|m_i| \in M_{x'}} \diamond_{h_{\mathsf{G}}(m_i)} h_{\mathsf{S}}(z(|m_i|)) \right) \to \boxtimes h_{\mathsf{S}}(\gamma_{x'}(z)), \tag{85}$$

for all $c_0 \in \mathsf{Im}\mathsf{C}(h_{\mathsf{L}})$. The restriction to configurations in $\mathsf{Im}\mathsf{C}(h_{\mathsf{L}})$ is not problematic, since the extension property only concerns the action of δ on local neighbourhood configurations z' where all cells in the image of h_{G} obtain a state in the image of h_{S} (i.e., extensions of $h_{\mathsf{S}} \circ z \circ (h_{\mathsf{G}}^{M_{x'}})^{-1}$ for some $z \colon X \to S_X$). All such neighbourhood configurations indeed arise as a subconfiguration in $\mathsf{Im}\mathsf{C}(h_{\mathsf{L}})$. Hence there exist enough choices for c_0 to use (85) to completely specify δ on inputs z'.

Since $h_{\mathsf{G}}[M] \subseteq N$ (Definition 8.13), it follows that $h_{\mathsf{G}}^{M_{x'}}[M_{x'}] \subseteq N_{h_{\mathsf{C}}(x')}$. Hence the premise of (85) contains a subset of the conjuncts of the premise of (84). Abstracting to propositional logic, we have two formulas:

$$(A_1 \wedge A_2 \wedge \cdots \wedge A_\ell) \to B_1,$$
$$(A_1 \wedge A_2 \wedge \cdots \wedge A_\ell \wedge C_1 \wedge \cdots \wedge C_q) \to B_2,$$

These can be further abstracted to

$$A \to B_1,$$
$$A \wedge C \to B_2,$$

Using the logical tautology $(A \to B_1) \to (A \wedge C \to B_1)$ and modus ponens (Lemmas 6.20 and 6.21) we find that $A \wedge C \to B_1$, i.e.,

$$\mathbb{Y}, c_0, h_{\mathsf{C}}(x') \vDash \left(\bigwedge_{|n| \in N_{h_{\mathsf{C}}(x')}} \diamond_n z'(|n|)) \right) \to \boxtimes h_{\mathsf{S}}(\gamma_{h_{\mathsf{C}}(x')}(z)). \tag{86}$$

So, given that the neighbours reachable from $h_{\mathsf{C}}(x')$ via paths $n \in h_{\mathsf{G}}[M]$ have states $z'(|n|_{h_{\mathsf{C}}(x')})$, the state of $h_{\mathsf{C}}(x')$ in the next timestep in the trace is $h_{\mathsf{S}}(\gamma_{h_{\mathsf{C}}(x')}(z))$. This holds regardless of the states z' assigns to cells at paths $N \setminus h_{\mathsf{G}}[M]$. Since $\delta_{h_{\mathsf{C}}(x')}$ also gives the next state of $h_{\mathsf{C}}(x')$, and since $h_{\mathsf{C}}(x')$ has at most one state per timestep, these states must coincide (i.e., semantically we know $B_1 \leftrightarrow B_2$) (i.e., varying the states of cells at paths $N \setminus h_{\mathsf{G}}[M]$ does not change the output of $\delta_{h_{\mathsf{C}}(x')}$ as long as the states of the paths in $h_{\mathsf{G}}[M]$ are given by z'). Hence the desired conclusion follows: $h_{\mathsf{S}}(\gamma_{x'}(z)) = \delta_{h_{\mathsf{C}}(x')}(z)$. □

[4]More precisely, we use Lemma 6.20 with the logical tautology $\alpha \wedge \beta \to \alpha$, with $\alpha := \diamond_{m_1} s_1 \to \diamond_{m_2} s_1$ and $\beta := \diamond_{m_2} s_1 \to \diamond_{m_1} s_1$. Now Eq. (82) gives $\alpha \wedge \beta$, thus by modus ponens (Lemma 6.21) we obtain α. Finally apply modus ponens on α, since the premise $\mathbb{X}, c'_0, x \vDash \diamond_{m_1} s_1$ is known to hold.

Corollary 8.23

If $x \leftrightsquigarrow y$ then $x \leftrightarrows y$.

Proof. Apply Lemma 8.22 twice. □

We will now prove the converse of Lemma 8.22 and Corollary 8.23, but this first requires an auxiliary lemma:

Lemma 8.24

Given $h\colon x \rightrightarrows y$, then for all $c_0 \in \mathsf{ImC}(h_\mathsf{L})$:

$$h_\mathsf{S} \circ G_\mathbb{X}(h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C}) \circ h_\mathsf{C}^{-1} = G_\mathbb{Y}(c_0) \restriction h_\mathsf{C}[\mathscr{M} \circledS x] \tag{87}$$

are the same configuration $h_\mathsf{C}[\mathscr{M} \circledS x] \to S_Y$.

$$\begin{array}{ccc} S_X^X & \xleftarrow{h_\mathsf{S}^{-1} \circ (-) \circ h_\mathsf{C}} & \mathsf{ImC}(h_\mathsf{L}) \\ {\scriptstyle G_\mathbb{X}}\downarrow & & \downarrow{\scriptstyle G_\mathbb{Y}} \\ S_X^X & \xrightarrow[h_\mathsf{S} \circ (-) \circ h_\mathsf{C}^{-1}]{} & \mathsf{ImC}(h_\mathsf{L}) \end{array} \tag{88}$$

It follows that

$$G_\mathbb{X}(h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C}) = h_\mathsf{S}^{-1} \circ G_\mathbb{Y}(c_0) \circ h_\mathsf{C} \tag{89}$$

(and that $h_\mathsf{S}^{-1} \circ G_\mathbb{Y}(c_0) \circ h_\mathsf{C}$ is well-defined).

Proof. Write h for h_C and take any $h_\mathsf{C}(x') \in h[\mathscr{M} \circledS x]$. Then:

$$\begin{aligned} \left(h_\mathsf{S} \circ G_\mathbb{X}(h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C}) \circ h_\mathsf{C}^{-1}\right)(h_\mathsf{C}(x')) &= h_\mathsf{S}(G_\mathbb{X}(h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C})(x')) \\ &= h_\mathsf{S}\left(\gamma_{x'}(\lambda\,|m| \in M_{x'}\,.\,(h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C})(m \circledS x'))\right) \\ &\qquad \text{// Definition global rule } G_\mathbb{X} \text{ (Eq. (28)).} \\ &= h_\mathsf{S}(\gamma_{x'}(z)) \quad \text{// Let } z := \lambda\,|m| \in M_{x'}\,.\,(h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C})(m \circledS x')\text{:} \\ &= \delta_{h_\mathsf{C}(x')}(z') \qquad \text{// Extension property (Definition 8.14.4).} \end{aligned}$$

for any extension $z'\colon Y \to S_Y$ of $h_\mathsf{S} \circ z \circ (h_\mathsf{G}^{M_{x'}})^{-1}\colon h_\mathsf{G}^{M_{x'}}[M_{x'}] \to S_Y$. The latter equals

$$\begin{aligned} h_\mathsf{S} \circ z \circ (h_\mathsf{G}^{M_{x'}})^{-1} &= \lambda\,|m| \in ((h_\mathsf{G}^{M_{x'}})^{-1} \circ h_\mathsf{G}^{M_{x'}})[M_{x'}]\,.\,(h_\mathsf{S} \circ h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C})(m \circledS x') \\ &= \lambda\,|m| \in M_{x'}\,.\,(c_0 \circ h_\mathsf{C})(m \circledS x') \\ &= \lambda\,|m| \in M_{x'}\,.\,c_0(h_\mathsf{G}(m) \circledS h_\mathsf{C}(x')) \\ &= \lambda\,|n| \in h_\mathsf{G}^{M_{x'}}[M_{x'}]\,.\,c_0(n \circledS h_\mathsf{C}(x')). \qquad \text{// Since } h_\mathsf{G}^{M_{x'}} \text{ is an injection.} \end{aligned}$$

The extension property guarantees that $\delta_{h_\mathsf{C}(x')}(z')$ behaves the same on any extension z' of (// Since $h_\mathsf{G}^{M_{x'}}$ is an injection.), including the intuitively natural extension:

$$z' := \lambda\,|n| \in N_{h_\mathsf{C}(x')}\,.\,c_0(n \circledS h_\mathsf{C}(x')).$$

Continuing where we left off in (// Extension property (Definition 8.14.4).) with this choice of z', we find:

$$\begin{aligned} &= \delta_{h_\mathsf{C}(x')}(z') \\ &= \delta_{h_\mathsf{C}(x')}\left(\lambda\,|n| \in N_{h_\mathsf{C}(x')}\,.\,c_0(n \circledS h_\mathsf{C}(x'))\text{.}\right) \\ &\overset{\text{def}}{=} G_\mathbb{Y}(c_0)(h_\mathsf{C}(x')), \qquad \text{// Definition global rule } G_\mathbb{Y} \text{ (Eq. (28)):} \end{aligned}$$

which was to be shown. □

Lemma 8.25

If $x \rightrightarrows y$ then $x \rightsquigarrow y$.

Proof. We need to prove that, for all $c_0 \in \mathsf{ImC}(h_\mathsf{L})$ it holds that

$$\mathbb{X}, h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C}, x \vDash \phi \text{ iff } \mathbb{Y}, c_0, y \vDash h_\mathsf{L}(\phi).$$

We will prove the stronger claim that for all $k_0 \in \mathsf{ImC}(h_\mathsf{L})$ and all $x' = m \otimes x \in \mathscr{M} \otimes x$:

$$\mathbb{X}, h_\mathsf{S}^{-1} \circ k_0 \circ h_\mathsf{C}, x' \vDash \phi \text{ iff } \mathbb{Y}, k_0, h_\mathsf{C}(x') \vDash h_\mathsf{L}(\phi). \tag{90}$$

For convenience, we omit explicitly notating the frames $\mathbb{X}$ and $\mathbb{Y}$.

Case $\phi = \boxtimes\psi$:
This is the only difficult case. Observe the following:

$$\begin{aligned}
&\bigforall_{c_0 \in \mathsf{ImC}(h_\mathsf{L})} h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C}, x' \vDash \boxtimes\psi \\
\text{iff } &\bigforall_{c_0} G_\mathbb{X}(h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C}), x' \vDash \psi && \text{// Semantics } \boxtimes. \\
\text{iff } &\bigforall_{c_0} h_\mathsf{S}^{-1} \circ G_\mathbb{Y}(c_0) \circ h_\mathsf{C}, x' \vDash \psi && \text{// Lemma 8.24 Eq. (89).} \\
\text{iff } &\bigforall_{c_0} G_\mathbb{Y}(c_0), h_\mathsf{C}(x') \vDash h_\mathsf{L}(\psi) && \text{// Induction hypothesis (with } k_0 := G_\mathbb{Y}(c_0)). \\
\text{iff } &\bigforall_{c_0} G_\mathbb{Y}(c_0), h_\mathsf{C}(x') \vDash \boxtimes h_\mathsf{L}(\psi) && \text{// Semantics } \boxtimes. \\
\text{iff } &\bigforall_{c_0} G_\mathbb{Y}(c_0), h_\mathsf{C}(x') \vDash h_\mathsf{L}(\boxtimes\psi).
\end{aligned}$$

Case $\phi = s \in S_X$ (case $\mathsf{v_S}$ is analogous):
Fix any arbitrary $c_0 \in \mathsf{ImC}(h_\mathsf{L})$, then:

$$\begin{aligned}
&\mathbb{X}, h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{L}, x' \vDash s \\
\text{iff } &h_\mathsf{S}^{-1}(c_0(h_\mathsf{C}(x'))) = s \\
\text{iff } &h_\mathsf{S}(h_\mathsf{S}^{-1}(c_0(h_\mathsf{C}(x')))) = h_\mathsf{S}(s) && \text{// } h_\mathsf{S} \text{ is an injection.} \\
\text{iff } &c_0(h_\mathsf{C}(x')) = h_\mathsf{S}(s) \\
\text{iff } &c_0, h_\mathsf{C}(x') \vDash h_\mathsf{S}(s),
\end{aligned}$$

and indeed $h_\mathsf{L}(\phi) = h_\mathsf{S}(s)$ in this case.

Case $\phi = \diamond_m\psi$ (for some $m \in \mathscr{M}$):
Again fix any arbitrary $c_0 \colon \mathsf{ImC}(h_\mathsf{L})$ and observe:

$$\begin{aligned}
&\mathbb{X}, h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C}, x' \vDash \diamond_m\psi \\
\text{iff } &\mathbb{X}, h_\mathsf{S}^{-1} \circ c_0 \circ h_\mathsf{C}, m \otimes x' \vDash \psi && \text{// Semantics } \diamond_m. \\
\text{iff } &\mathbb{Y}, c_0, h_\mathsf{C}(m \otimes x') \vDash h_\mathsf{L}(\psi) && \text{// Induction hypothesis.} \\
\text{iff } &\mathbb{Y}, c_0, h_\mathsf{G}(m) \mathbin{⦾} h_\mathsf{C}(x') \vDash h_\mathsf{L}(\psi) \\
\text{iff } &\mathbb{Y}, c_0, h_\mathsf{C}(x') \vDash \diamond_{h_\mathsf{G}(m)} h_\mathsf{L}(\psi) && \text{// Semantics } \diamond_{h_\mathsf{G}(m)}. \\
\text{iff } &\mathbb{Y}, c_0, h_\mathsf{C}(x') \vDash h_\mathsf{L}(\diamond_m\psi).
\end{aligned}$$

Remaining cases:
The $\phi = \bot$ case is trivial. The $\phi = \neg\psi$, $\phi = \psi_1 \vee \psi_2$ and $\phi = \bigvee_{i\in I} \psi_i$ cases for almost immediately from the induction hypothesis. □

Corollary 8.26

If $x \leftrightarroweq y$ then $x \leftrightsquigarrow y$.

Proof. Apply Lemma 8.25 twice. □

We now summarise the work done above as:

Theorem 8.27

$$x \rightrightarrows y \quad \text{iff} \quad x \rightsquigarrow y$$

and

$$x \leftrightarroweq y \quad \text{iff} \quad x \leftrightsquigarrow y$$

(given and $\mathsf{card}(S_X) \geq 2$ and $\mathsf{card}(S_Y) \geq 2$, and in case of $\mathcal{L}_{\mathsf{stg}}$, also that M and N are finite).

8.3.4 Example applications

The following example shows a mutual similarity between two CA on very different graphs. It is based on an example from the literature [36, Prop. 2, 2nd proof], and demonstrates how the current approach, which separated the set of cells and the paths monoid, fits well with the description of the simulation:

Example 8.28 (Hexagonal to 3D)**.** Let the hexagonal paths grid $\mathscr{H}$ be the Abelian group generated over the set $\{A, B, C\}$ with equalities $ABC = \mathbf{1}$, $AB = BA$, $AC = CA$ and $BC = CB$[5].

As neighbourhood we use $M = \{\mathbf{1}, A, B, C, A^{-1}, B^{-1}, C^{-1}\}$. Since this group is Abelian, we can draw it as the following grid:

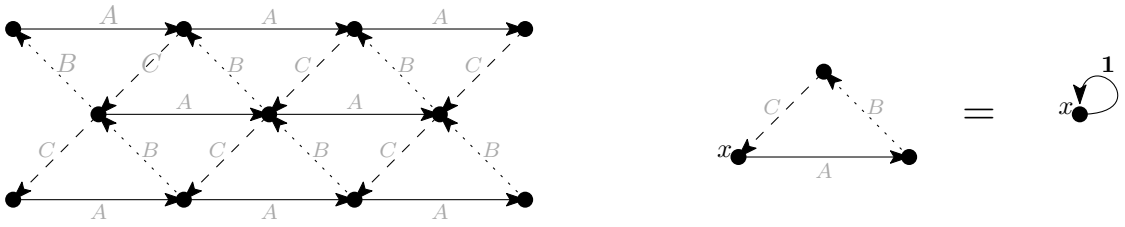


Figure 21: Right: a piece of the grid encoded by $\mathscr{H}$, not showing the inverse group elements. Right: the equality $ABC = \mathbf{1}$.

As cells of the CA frame $\mathbb{X} = (\mathscr{H}, N, S, X, \langle\chi, \gamma\rangle)$ we take $X = \mathscr{H}$, and the neighbours-mapping $\circledast$ is the multiplication on $\mathscr{H}$. The state set S and local rule $\gamma\colon (M_X \to S)^X$ can be arbitrary (as long as $\mathsf{card}(S) \geq 2$). Since γ_x is the same for all $x \in X$, we can simply write $\gamma_x = \gamma_*$. Since all M neighbours of every cell are distinct, we will (with a minor above of notation) also write M for M_x.

[5]The equalities $AC = CA$ and $BC = CB$ are superfluous and can be derived from the pair $ABC = \mathbf{1}$ and $AB = BA$.

We can simulate this CA on $\mathbb{Z}^3$ by using the CA

$$\mathbb{Y} = (\mathbb{Z}^3, N = \{(i,j,k) \mid |i| + |j| + |k| \leq 1\}, S, Y = \mathbb{Z}^3 + (1,1,1)\mathbb{Z}^3, \langle \xi, \delta \rangle),$$

such that $\circledS$ is addition modulo $(1,1,1)$ (so $(a,b,c) \circledS ((p,q,r)+(1,1,1)\mathbb{Z}^3) \stackrel{\text{def}}{=} (p+a, q+b, r+c)+(1,1,1)\mathbb{Z}^3$). Again we will write δ_* for $\delta_{\vec{y}}$ and N for $N_{\vec{y}}$. It remains to define δ, but this is easier after a few more observations.

The intuitive layout of Y in $\mathbb{Z}^3$ is as follows: if we fix the value 0 for the third coordinate, we obtain a plane containing every cell of Y exactly once (for every $(i,j,k) \in \mathbb{Z}^3 + (1,1,1)\mathbb{Z}^3$, we have that $(i-k, j-k, 0) + (1,1,1)\mathbb{Z}^3$ is the same cell, and the only occurrence of this cell at third coordinate 0). The plane where the third coordinate is 1 is the same, but with the origin at $(1,1,1)$ instead of $(0,0,0)$. We see that every cell is repeated endlessly along diagonal lines with slope $(1,1,1)$.

To define δ, observe that every element of $\mathscr{H}$ can be written as $A^i B^j C^k$ because $\mathscr{H}$ is Abelian. Since $ABC = \mathbf{1}$, we may assume a canonical representation of the cells in $\mathscr{H}$ (and hence X) in which at least one of i, j and k is zero. This already gives an isomorphism from $\mathscr{H}$ to $\mathbb{Z}^3 + (1,1,1)\mathbb{Z}^3$:

$$h \colon A^i B^j C^k \mapsto (i,j,k).$$

Intuitively, we identify A, B and C with the axes of the 3D grid. The A and A^{-1} neighbours become the left and right neighbours, the B and B^{-1} the forth and back neighbours, and C and C^{-1} the up and down neighbours. Thus $h \colon M \cong N$, and we define $\delta_* \stackrel{\text{def}}{=} \gamma_* \circ h^M_{\mathsf{G}} \colon S^N \to S$.

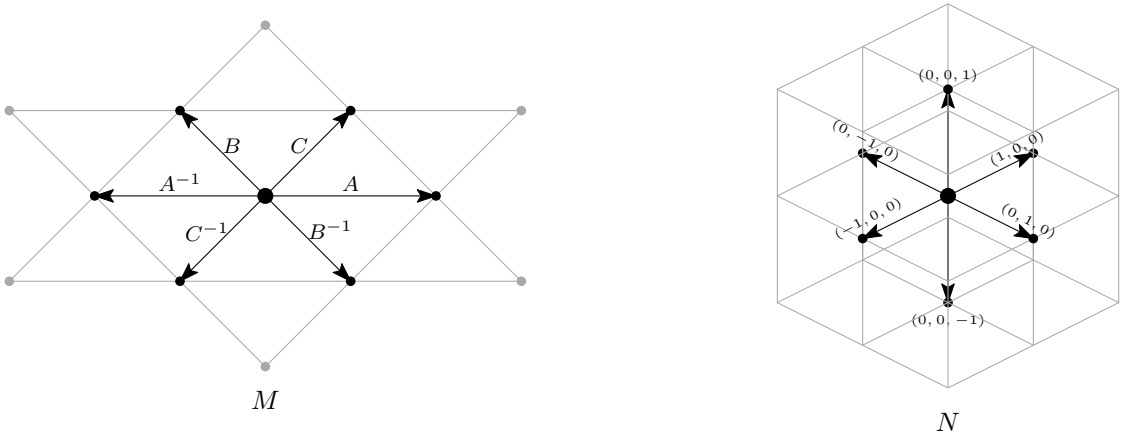


Figure 22: Left: the neighbourhood M on the hexagonal grid. Right: the isomorphic neighbourhood N in $\mathbb{Z}^3$.

After taking $h_{\mathsf{S}} \stackrel{\text{def}}{=} \mathsf{Id}_S$, we see that h trivially satisfies extension property of a general CA morphism (Definition 8.14.4), and the other requirements are also straightforward to check. Since h also has an inverse, we obtain a mutual similarity $A^i B^j C^k \leftrightarrows (i,j,k) + (1,1,1)\mathbb{Z}^3$ for every $(i,j,k) \in \mathbb{Z}^3$. Furthermore, from Theorem 8.27 it follows that (up to an isomorphism) $\mathsf{Log}(A^i B^j C^k)$ and $\mathsf{Log}((i,j,k))$ are the same.

The Hennessy-Milner theorems can also be used to prove what kind of simulations are impossible. For example, one cannot simulate a CA with a quiescent state on an CA without a quiescent state:

Lemma 8.29

Consider frames $\mathbb{X}$ and $\mathbb{Y}$ such that $\mathbb{X}$ has a quiescent state. If $x \rightrightarrows y$ for some $x \in X$ and $y \in Y$, then $\mathbb{Y}$ has a quiescent state.

Proof. Let ψ be the has-a-quiescent-state-formula (Eq. (56)), then $\mathbb{X}, x \vDash \psi$. By Lemma 8.25 it holds that

$$\mathbb{Y}, c_0, y \vDash h_{\mathsf{L}}(\psi) \tag{91}$$

for all $c_0 \colon Y \to h_{\mathsf{S}}[S_X]$. Now it might be that $h_{\mathsf{C}}[M \mathbin{\textcircled{x}} x] \subsetneq N \mathbin{\textcircled{x}} y$, thus (91) does not completely specify δ_y on the constant neighbourhood configuration $\mathsf{const}_{h_{\mathsf{S}}(s)} \colon (N \mathbin{\textcircled{x}} y) \to S_Y$ yet. This is not an issue, since we can pick a c_0 such that all neighbours of y (even those not in the image of X under h_{C}) have state $h_{\mathsf{S}}(s)$, in which case (91) does imply that $\delta_y(\mathsf{const}_{h_{\mathsf{S}}(s)}) = h_{\mathsf{S}}(s)$. The latter is exactly the quiescence condition. □

The converse may not hold when h_{S} is not surjective; in this case $\mathbb{Y}$ might have a quiescent state in $S_Y \setminus h_{\mathsf{S}}[S_X]$ which is not of a concern in the definition of $x \rightsquigarrow y$.

Finally, the Hennessy-Milner theorem allows to prove that certain properties cannot be axiomatised in $\mathcal{L}'_{\mathsf{CA}}$. One example is the property of being an OCA (Definition 7.23 of Section 7.3).

Lemma 8.30

There exists no $\psi \in \mathcal{L}'_{\mathsf{CA}}$ such that $\mathbb{X}, x \vDash \psi$ iff there exist no nontrivial path from x to x in $\mathfrak{G}^N$.

Proof. Let $\mathbb{X} = (\mathbb{Z}, \{0\}, S, \mathbb{Z}, \langle \chi, \gamma \rangle)$ and $\mathbb{Y} = (\mathbb{Z}, \{-1, 0, 1\}, S, \mathbb{Z}, \langle \chi, \delta \rangle)$ where $S = \{\bullet, \circ\}$, $\gamma_z = \mathsf{const}_{\circ} \colon \{z\} \to S$ and $\delta_z = \mathsf{const}_{\circ} \colon \{z-1, z, z+1\} \to S$ for all $z \in \mathbb{Z}$. Using $h_{\mathsf{C}} = \mathsf{Id}_{\mathbb{Z}}$, $h_{\mathsf{S}} = \mathsf{Id}_S$ and $h_{\mathsf{G}} = \mathsf{Id}_{\mathbb{Z}}$ we obtain a mutual similarity $h \colon z \rightrightarrows z$ for all $z \in \mathbb{Z}$ (between z in $\mathbb{X}$ to z in $\mathbb{Y}$). Thus by Theorem 8.27 also $z \rightsquigarrow z$, since $\mathsf{ImC}(h_{\mathsf{L}}) = S^{\mathbb{Z}}$ are *all* configurations on $\mathbb{Y}$, this implies $\mathbb{X}, z \vDash \psi$ iff $\mathbb{Y}, z \vDash \psi$.

Now suppose there exists a ψ that is validated in a point z iff there exist no nontrivial path in $\mathfrak{G}^N$ from z to z. But $\mathbb{X}$ is an OCA, so $\mathbb{X}, z \vDash \psi$ holds for all $z \in \mathbb{Z}$. Hence by the previous observation, also $\mathbb{Y}, z \vDash \psi$ for all $z \in \mathbb{Z}$. The latter implies that $\mathbb{Y}$ is an OCA; but this is a contradiction since $\mathbb{Y}$'s information graph is the essentially undirected (c.f. Definition 7.19) Cayley graph of $\mathbb{Z}$. □

8.4 Summary of the Hennessy-Milner theorem

This section proved a Hennessey-Milner theorem: a direct link between the logic of CA and the ability of CA to semantically simulate each other. We first saw, in Theorem 8.12, that CA $\mathcal{X}$ and $\mathcal{Y}$ defined over the same signature $(\mathscr{M}, N, S)$ validate exactly the same logic iff there exist coalgebra homomorphisms $k \colon \mathcal{X} \to \mathcal{Y}$ and $h \colon \mathcal{Y} \to \mathcal{X}$.

Thereafter we generalised coalgebra homomorphisms to *general CA morphisms* between frames. In order to translate signatures, these CA morphisms are coalgebra homomorphisms between cells extended with an additional monoid homomorphism and a state-set map. Intuitively, exists a general CA morphism $h \colon \mathbb{X} \to \mathbb{Y}$ when $\mathbb{X}$ can be simulated by $\mathbb{Y}$.

We likewise generalised *logical equivalence* to *logical simulation* in order to compare logics of frames on different signatures. This is implemented by allowing a monoid homomorphism and state-set map to translate the signature-dependent symbols in the logic.

Finally, we proved a generalised Hennessy-Milner theorem (Theorem 8.27): frames can simulate each other via general CA morphisms if and only if their logics can be embedded into one each other's.

9 Concluding remarks

We finish this thesis by reflecting on the advantages and disadvantages of the presented constructions, outlining directions for potential future work, and a final summary.

9.1 Discussion of the coalgebaic model

The main particularity of the present coalgebraic model of CA is the separation between the set of cells X and the monoid $\mathscr{M}$ of paths between cells; most existing works identify those sets. The separation seems to offer new flexibility. For example, we showed in Example 8.28 how to embed the hexagonal grid into $\mathbb{Z}^3$, which was straightforward since we only use the cells of the hexagonal CA and adjust the paths accordingly (intuitively, we identify multiple grid points with the same cell). This should be compared to the non-coalgebraic description by Roká [36], which had to explain how to translate a configuration on the hexagonal grid to a configuration on $\mathbb{Z}^3$ by copying the state of a source cell to all $\mathbb{Z}^3$ grid-points with which it is identified.

Classes of general CA We proposed several classes of general CA with different levels of uniformity between local rules in Section 4.1, None of these is the obvious "correct" class: this is probably application dependent. However, our coalgebras are sufficiently expressive that any coalgebraic categories of CA according to of the desired degree of uniformity simply arise as full subcategories of $\mathbf{CCA}_{\mathsf{G}}$, the category of general CA.

The standard coalgebra framework The present model does not fit in well with existing approaches of describing dynamical systems coalgebraically. The strongest discrepancies are Section 5, which showed that the concept of *behavioural equivalence* for our coalgebras has little to do with the actual behaviour of CA. But the issue runs deeper: traditionally, the elements of carrier sets of coalgebras are called "*states*". In our model, this are the *cells*, and part of the static structure rather than the dynamics of the system. The question whether there exist a coalgebraic model for CA that is better aligned with existing work remains open for future work.

Alternative base categories Our coalgebraic model uses an endofunctor on **Set**, but future work could investigate other base categories with more structure. For example, we can "lift" the functor C_{G} to an endofunctor on the category of posets, in such a way that C_{G} is recovered by precomposing with the forgetful functor. This would result in orderings on cells and states, and monotone general CA morphisms and local rules. Alternatively, we can consider the base category of small categories and use the morphisms of a category as connections between cells, instead of a separate monoid.

9.2 Discussion of the new modal logic

The presented modal logic $\mathcal{L}_{\mathsf{CA}}$ is quite expressive; aside from any finite spatio-temporal pattern in the trace of a CA, it can sometimes even describe asymptotic behaviour of the trace (recall Examples 7.13 and Lemma 7.14). However, infinitary disjunctions come at the cost of making the logic undecidable (Corollary 7.15). We conjecture that $\mathcal{L}_{\mathsf{stg}}$ is decidable, at least for CA with a finite state sets and neighbourhoods, but a proof of this remains open for future work.

One-way cellular automata Lemma 8.30 shows that $\mathcal{L}'_{\mathsf{CA}}$ does not have a formula that expresses that a CA is a one-way CA. This is not in conflict with the design goal of the language, since the property of

being an OCA is not visible in the trace. Indeed, the proof of Lemma 8.30 constructs two CA with the same traces, but one is an OCA and the other not.

This example shows that it is debatable whether the definition of an OCA is "correct", and more generally, whether the definition of the information flow graph (Definition 7.18) is "correct". The alternative is to define one-directedness based on the actual dynamics of the CA: we could for example define *essential one-directnedness* as follows: a CA is essentially one-directed if for any cell x in a configuration c_t of which it is known that x's state in configuration c_{t+1} is s_{t+1}, for any $k \in \mathbb{N}$, the state of x in c_{t+1+k} is the same for any state of x in the current configuration c_t. While somewhat technical, this expresses that information about x's past states cannot flow back to influence its future states (other than directly influence the direct successor state of course). One could generalise the definition of the information flow graph analogously. While semantically more intuitive, it may not solve the inexpressivity issue: we have not found a formula to express *essential one-directedness*.

A more general definition of simulation? Theorem 8.27, our generalised Hennessy-Milner theorem, is based on our definition of a *simulation* and CA being *mutually similar* (Definition 8.15). It is straightforward to see that a simulation $h\colon x \rightrightarrows y$ indeed embeds the dynamics of $\mathbb{X}$ visible from x intro y, but note that this is a simulation that always simulated 1 cell by 1 cell, 1 state by 1 state, and 1 timestep by 1 timestep. There are many examples of more general definitions of simulations between CA found in the literature, such as allowing groups of cell to simulate groups of cells, and allowing a constant time delay between the simulated and simulating CA, or even mapping entire configurations to configurations (see e.g., [36] for simulations of CA on Cayley graphs, and [29, §5.2] for a very general definition for graphs of the form $\mathbb{Z}^d$). It seems possible to generalise our definition of $\rightrightarrows$ accordingly, and it might also be possible to extend the definition of logical equivalence with a search-and-replace operation that multiplies occurrences of $\boxtimes$ (to express a delay), replaces occurrences of atomic states by a conjunction of states (when a single state of the simulated CA can be represented by multiple states in the simulating CA) or even replaces occurrences of atomic states $s \in S$ by spatially arranges "blocks" of states (e.g., when s is encoded in the simulating CA by multiple cells in the states $s_1, \dots, s_k$ arranged in the pattern given by $m_1, \dots, m_k$ then one would replace s by $\Diamond_{m_1} s_1 \wedge \Diamond_{m_2} s_2 \wedge \dots \wedge \Diamond_{m_k} s_k$). This possibility remains open for future work.

Uniform substitution We showed in Section 6.3 that the global variables of out logic, which semantically represent the states in the initial configuration, allow a limited form of uniform substitution: only substitution of time-invariant formulas. In future work we may investigate the alternative option of assigning propositional letters to cells (as in the Kripke semantics), in which case full uniform substitution likely becomes possible.

9.3 Conclusion

We have successfully constructed a modal logic for describing spatio-temporal patterns in the trace of a CA (see Example 7.1). Aside from these patterns, the logic can also express some global properties of a CA, such as nilpotency (Example 7.13) and periodicity (Example 7.14). To implement the semantics of the logic, we constructed a novel coalgebraic model of CA that separates cells from paths between cells. We proposed two variants of a a generalisation of CA in which different cells can have different local behaviour. Our coalgebraic model is sufficiently flexible to also represent both *irregular* and *non-uniform* CA, to which the logic also applies.

The logic is not based on the predicate-lifting framework for coalgebraic modal logic, since we showed that this framework cannot express dynamic properties for our coalgebraic representation of CA.

The syntax of the logic is uniform for all CA, but formulas may contain state labels or paths labels specific to a CA. However, we gave a generalised Hennessy-Milner theorem stating that the formulas validated by a CA can be translated into formulas validated by another CA if and only if the latter CA can directly simulate the

former. This result simplifies to a direct logical equivalence in the special case where both CA have exactly the same states and paths.

References

[1] Kamalika Bhattacharjee, Nazma Naskar, Souvik Roy, and Sukanta Das.
"A survey of cellular automata: types, dynamics, non-uniformity and applications".
In: *Natural Computing* 19.2 (July 2018), pp. 433–461. ISSN: 1572-9796.
DOI: 10.1007/s11047-018-9696-8. URL: http://dx.doi.org/10.1007/s11047-018-9696-8.

[2] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*.
Cambridge University Press, 2001. URL: www.cambridge.org/9780521527149.

[3] Dave Burraston, Ernest A. Edmonds, Dan Livingstone, and Eduardo Reck Miranda.
"Cellular Automata in MIDI based Computer Music".
In: *International Conference on Mathematics and Computing*. 2004.
URL: http://hdl.handle.net/2027/spo.bbp2372.2004.047.

[4] Silvio Capobianco and Tarmo Uustalu. "A categorical outlook on cellular automata".
In: *Journées Automates Cellulaires* (2010).

[5] Silvio Capobianco and Tarmo Uustalu. "Additive Cellular Automata Graded-Monadically".
In: *International Symposium on Principles and Practice of Declarative Programming (PPDP 2023)* (2023).

[6] Corina Cîrstea, Alexander Kurz, Dirk Pattinson, Lutz Schröder, and Yde Venema.
"Modal Logics are Coalgebraic1". In: *The Computer Journal* 54.1 (Feb. 2009), pp. 31–41.
ISSN: 0010-4620. DOI: 10.1093/comjnl/bxp004.
eprint: https://academic.oup.com/comjnl/article-pdf/54/1/31/963631/bxp004.pdf.
URL: https://doi.org/10.1093/comjnl/bxp004.

[7] Stathis Delivorias, Haralampos Hatzikirou, Rafael Peñaloza, and Dirk Walther.
"Detecting Emergent Phenomena in Cellular Automata Using Temporal Description Logics".
In: *Cellular Automata*. Ed. by Jarosaw Ws, Georgios Ch. Sirakoulis, and Stefania Bandini.
Cham: Springer International Publishing, 2014, pp. 357–366. ISBN: 978-3-319-11520-7.

[8] M. Delorme and J. Mazoyer. *Cellular Automata A Parallel Model*.
Springer Science+Business Media Dordrecht, 1999.

[9] Bruno Durand and Zsuzsanna Róka. "The game of life: universality revisited".
In: *Cellular Automata A Parallel Model*. Springer Science+Business Media Dordrecht, 1999.

[10] G. Bard Ermentrout and Leah Edelstein-Keshet.
"Cellular Automata Approaches to Biological Modeling".
In: *Journal of Theoretical Biology* 160.1 (1993), pp. 97–133. ISSN: 0022-5193.
DOI: https://doi.org/10.1006/jtbi.1993.1007.
URL: https://www.sciencedirect.com/science/article/pii/S0022519383710076.

[11] Martin Gardner. "MATHEMATICAL GAMES". In: *Scientific American* 223.4 (1970), pp. 120–123.
ISSN: 00368733, 19467087. URL: http://www.jstor.org/stable/24927642 (visited on 06/27/2024).

[12] Matthew Hennessy and Robin Milner. "Algebraic laws for nondeterminism and concurrency".
In: *J. ACM* 32.1 (Jan. 1985), pp. 137–161. ISSN: 0004-5411. DOI: 10.1145/2455.2460.
URL: https://doi.org/10.1145/2455.2460.

[13] Jesse Hughes and Bart Jacobs. "Simulations in coalgebra".
In: *Theoretical Computer Science* 327.1 (2004). Selected Papers of CMCS '03, pp. 71–108.
ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2004.07.022.
URL: https://www.sciencedirect.com/science/article/pii/S030439750400444X.

[14] Navot Israeli and Nigel Goldenfeld.
"Computational Irreducibility and the Predictability of Complex Physical Systems".
In: *Physical Review Letters* 92.7 (Feb. 2004). DOI: 10.1103/physrevlett.92.074105.
URL: https://doi.org/10.1103%2Fphysrevlett.92.074105.

[15] Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*.
Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2016.

[16] Bart Jacobs and Jan Rutten. "An introduction to (co)algebra and (co)induction". In:
Advanced Topics in Bisimulation and Coinduction. Ed. by Davide Sangiorgi and Jan Rutten.
Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2011, pp. 38–99.

[17] Lawrence Johnson, Georgios N. Yannakakis, and Julian Togelius.
"Cellular automata for real-time generation of infinite cave levels".
In: *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. PCGames '10.
Monterey, California: Association for Computing Machinery, 2010. ISBN: 9781450300230.
DOI: 10.1145/1814256.1814266. URL: https://doi.org/10.1145/1814256.1814266.

[18] André Joyal, Mogens Nielsen, and Glynn Winskel. "Bisimulation from Open Maps".
In: *Information and Computation* 127 (1996).

[19] Jarkko Kari. "The Nilpotency Problem of One-Dimensional Cellular Automata".
In: *SIAM Journal on Computing* 21.3 (1992), pp. 571–586. DOI: 10.1137/0221036.
eprint: https://doi.org/10.1137/0221036. URL: https://doi.org/10.1137/0221036.

[20] Jarkko Kari. "Theory of cellular automata: A survey".
In: *Theoretical Computer Science* 334.1 (2005), pp. 3–33. ISSN: 0304-3975.
DOI: https://doi.org/10.1016/j.tcs.2004.11.021.
URL: https://www.sciencedirect.com/science/article/pii/S030439750500054X.

[21] Jarkko Kari and Nicolas Ollinger. "Periodicity and Immortality in Reversible Computing".
In: *Mathematical Foundations of Computer Science 2008*.
Ed. by Jerzy Ochmaski Edwardand Tyszkiewicz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008,
pp. 419–430. ISBN: 978-3-540-85238-4.

[22] Bartek Klin. "Coalgebraic Modal Logic Beyond Sets".
In: *Electronic Notes in Theoretical Computer Science* (173 2007).

[23] Clemens Kupke and Dirk Pattinson. "Coalgebraic semantics of modal logics: An overview".
In: *Theoretical Computer Science* 412.38 (2011). CMCS Tenth Anniversary Meeting, pp. 5070–5094.
ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2011.04.023.
URL: https://www.sciencedirect.com/science/article/pii/S0304397511003215.

[24] Petr Krka. "On topological dynamics of Turing machines".
In: *Theoretical Computer Science* 174.1 (1997), pp. 203–216. ISSN: 0304-3975.
DOI: https://doi.org/10.1016/S0304-3975(96)00025-4.
URL: https://www.sciencedirect.com/science/article/pii/S0304397596000254.

[25] Tom Leinster. *Basic Category Theory*. Vol. 143. Cambridge Studies in Advanced Mathematics.
Cambridge University Press, July 2014. URL: http://www.cambridge.org/9781107044241.

[26] Saunders MacLane. *Categories for the Working Mathematician*.
Graduate Texts in Mathematics, Vol. 5. New York: Springer-Verlag, 1971, pp. ix+262.

[27] Surendra Kumar Nanda, Suneeta Mohanty, and Prasant Kumar Pattnaik. "A Survey on Cellular Automata-Based Security Systems for Information Security and Optimized Performance". In: *Proceedings of 2nd International Conference on Smart Computing and Cyber Security*. Ed. by Prasant Kumar Pattnaik, Mangal Sain, and Ahmed A. Al-Absi. Singapore: Springer Nature Singapore, 2022, pp. 235–242. ISBN: 978-981-16-9480-6.

[28] S. Nandi, B.K. Kar, and P. Pal Chaudhuri. "Theory and applications of cellular automata in cryptography". In: *IEEE Transactions on Computers* 43.12 (1994), pp. 1346–1357. DOI: 10.1109/12.338094.

[29] Nicolas Ollinger. "Universalities in Cellular Automata". In: Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok. *Handbook of Natural Computation*. Springer-Verlag Berlin Heidelberg, 2012.

[30] Dirk Pattinson. "Coalgebraic modal logic: soundness, completeness and decidability of local consequence". In: *Theoretical Computer Science* 309 (2003). DOI: 10.1016/S0304-3975(03)00201-9.

[31] Dirk Pattinson. "Semantical Principles in the Modal Logic of Coalgebras". In: *STACS 2001*. Ed. by Afonso Ferreira and Horst Reichel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 514–526. ISBN: 978-3-540-44693-4.

[32] *Physica D: Nonlinear Phenomena*. Vol. 10. 1984. URL: https://www.sciencedirect.com/journal/physica-d-nonlinear-phenomena/vol/10/issue/1.

[33] *Physica D: Nonlinear Phenomena*. Vol. 45. 1990. URL: https://www.sciencedirect.com/journal/physica-d-nonlinear-phenomena/vol/45/issue/1.

[34] Dan (sigfpe) Piponi. *Evaluating cellular automata is comonadic*. Accessed 4 July 2024. 2006. URL: http://blog.sigfpe.com/2006/12/evaluating-cellular-automata-is.html.

[35] Emily Riehl. *Category Theory in Context*. Dover, 2016. URL: http://www.math.jhu.edu/%5C~%7B%7Deriehl/context.pdf.

[36] Zsuzsanna Róka. "Simulations between cellular automata on Cayley graphs". In: *Theoretical Computer Science* 225 (1999).

[37] Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok. *Handbook of Natural Computation*. Springer-Verlag Berlin Heidelberg, 2012.

[38] Jan J. M. M. Rutten. "The Method of Coalgebra: exercises in coinduction". In: *World Congress on Formal Methods*. 2019. URL: https://api.semanticscholar.org/CorpusID:86847919.

[39] Jan J.M.M. Rutten. "Universal coalgebra: a theory of systems". In: *Theoretical Computer Science* 249.1 (2000). Modern Algebra, pp. 3–80. ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(00)00056-6. URL: https://www.sciencedirect.com/science/article/pii/S0304397500000566.

[40] Davide Sangiorgi. *Introduction to Bisimulation and Coinduction*. Cambridge University Press, 2011.

[41] Lutz Schröder. "Expressivity of coalgebraic modal logic: The limits and beyond". In: *Theoretical Computer Science* (390 2008).

[42] Alvy Ray Smith. "Simple Computation-Universal Cellular Spaces". In: *J. ACM* 18.3 (July 1971), pp. 339–353. ISSN: 0004-5411. DOI: 10.1145/321650.321652. URL: https://doi.org/10.1145/321650.321652.

[43] Sam Staton. "Relating coalgebraic notions of bisimulation". In: *Logical Methods in Computer Science* 7.(1:13) (2011). DOI: 10.2168/LMCS-7. URL: https://doi.org/10.2168/LMCS-7.

[44] Penelope Sweetser and Janet Wiles.
"Combining Influence Maps and Cellular Automata for Reactive Game Agents".
In: *Intelligent Data Engineering and Automated Learning - IDEAL 2005*.
Ed. by Marcus Gallagher, James P. Hogan, and Frederic Maire.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 524–531. ISBN: 978-3-540-31693-0.

[45] Véronique Terrier. "Language Recognition by Cellular Automata".
In: Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok. *Handbook of Natural Computation*.
Springer-Verlag Berlin Heidelberg, 2012.

[46] Oleksii O Vodka and Mariia I Shapovalova. "Exploration of cellular autamata: a comprehensive review of dynamic modeling across biology, computer and materials science".
In: *Computer Methods in Materials Science* 23.4 (2023), p. 58.
URL: https://doi.org/10.7494/cmms.2023.4.0820.

[47] Gabriel Wainer, Qi Liu, Olivier Dalle, and Bernard P. Zeigler.
"Applying Cellular Automata and DEVS Methodologies to Digital Games: A Survey".
In: *Simulation & Gaming* 41.6 (2010), pp. 796–823. DOI: 10.1177/1046878110378708.
eprint: https://doi.org/10.1177/1046878110378708.
URL: https://doi.org/10.1177/1046878110378708.

[48] Baltasar Trancón y Widemann and Michael Hauhs.
"Distributive-Law Semantics for Cellular Automata and Agent-Based Models".
In: *Algebra and Coalgebra in Computer Science*.
Ed. by Andrea Corradini, Bartek Klin, and Corina Cîrstea. Springer-Verlag Berlin Heidelberg, 2011.
DOI: 10.1007/978-3-642-22944-2.

[49] Stephen Wolfram. "Cryptography with Cellular Automata".
In: *Advances in Cryptology — CRYPTO '85 Proceedings*. Ed. by Hugh C. Williams.
Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 429–432. ISBN: 978-3-540-39799-1.

[50] Stephen Wolfram. "Statistical mechanics of cellular automata".
In: *Rev. Mod. Phys.* 55 (3 July 1983), pp. 601–644. DOI: 10.1103/RevModPhys.55.601.
URL: https://link.aps.org/doi/10.1103/RevModPhys.55.601.