# 6 Chapter 6: Global Simulation of Cellular Automata

In the previous chapters, we studied several ways of comparing dynamical systems, in particular through subsystems, factors, and embeddings. In this chapter, we revisit these ideas from a computational perspective. Our goal now is to understand when the dynamics of one system should count as carrying out the *computation* of another. This immediately leads to an interesting notion, namely that of *simulation*.

The need for such a notion is already implicit in the theory of universal Turing machines. A universal Turing machine is said to simulate an arbitrary Turing machine; a modern computer is said to simulate a fluid; a neural circuit in the brain of a frog is said to simulate the motion of a fly passing by. Yet, despite its foundational role, the notion of simulation is rarely isolated and defined on its own. Usually one simply presents a particular construction of a simulation and verifies that it captures certain dynamical features of the system that it is purported to be simulating. This ambiguity is harmless in most circumstances, but is pressing when there are less obvious notions of simulation. How would we know, for instance, whether a cellular automaton is *performing computation*? We would not want to claim that a cellular automaton is performing a complex computation when in fact it is not, and the would-be computation is merely an artifact of a convoluted description of the system.

The first point to emphasize is that simulation is not the same thing as exact equality of dynamics. If one system is to simulate another, their state spaces need not be the same, and the simulated system need not proceed at exactly the same speed. Rather, one should expect the presence of a *translation layer*: one has to encode states of the simulated system into states of the simulator, let the simulator evolve for a suitable amount of time, and then decode the resulting state back, see Figure 1.

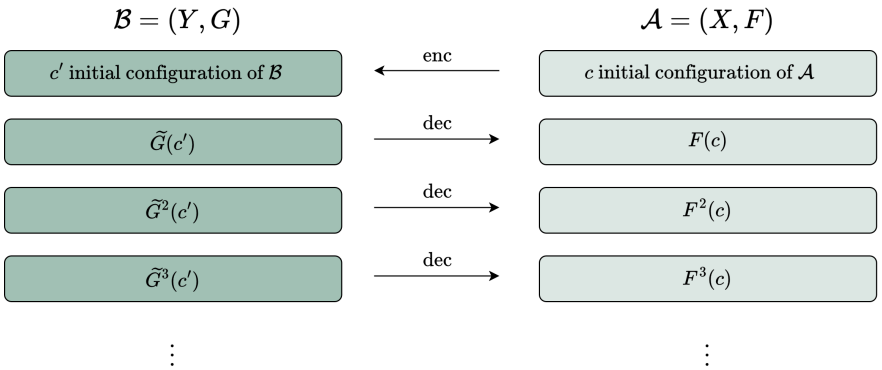


Figure 1: Intuitive illustration of a dynamical system $\mathcal{B}$ simulating $\mathcal{A}$, where enc, dec are some "computationally reasonable" maps, and $\tilde{G}$ is a simple transformation of $G$ which accounts, for instance, for a suitable time-delay (e.g., $\tilde{G} = G^\tau$ for some $\tau \in \mathbb{N}$).

This suggests that simulation should involve three pieces of extra data:

1. An *encoder*, translating configurations of the simulated system into configurations of the simulator,
2. A *decoder*, translating configurations of the simulator back into configurations of the simulated system, and
3. A *slowdown function*, specifying how many steps of the simulator correspond to one step of the simulated system.

The need for the slowdown function is important since a simulation does not, in general, reproduce one time step of the system being simulated in exactly one step of its own. The simulation may be slower, and the amount of slowdown may depend on the currently represented configuration.

There is, however, a further subtlety. If the encoder and decoder are allowed to be arbitrarily complicated, then the notion of simulation becomes almost empty. Indeed, one can hide the entire computation in the encoder or decoder and let the purported simulator do almost nothing at all. In such a case, it would be misleading to say that the simulator carries out the computation. The real computational work would have been off-loaded to the translation layer. Therefore, simulation must always be understood relative to some class of *admissible* or *constrained* translation procedures. In the present setting, the most natural way to express admissibility is by placing complexity bounds on the encoder and decoder themselves.

In this part of the course, we will focus on the computational capacity of cellular automata by studying what kinds of other systems a given cellular automaton $\mathcal{B}$ can simulate. To this end, we will distinguish two cases of simulation according to the kinds of systems simulated by $\mathcal{B}$.

1. Local simulation: $\mathcal{B}$ embeds a Turing machine within some bounded active region (possibly growing in time) of the grid; see Figure 2(a).

2. Global simulation: the laws of physics of $\mathcal{B}$ can simulate, in their entirety, the laws of physics of a different system, namely another CA $\mathcal{A}$. That is, any space-time diagram of $\mathcal{A}$ on an *infinite grid* can be recovered through computationally feasible maps from a suitable space-time diagram of $\mathcal{B}$; see Figure 2(b).

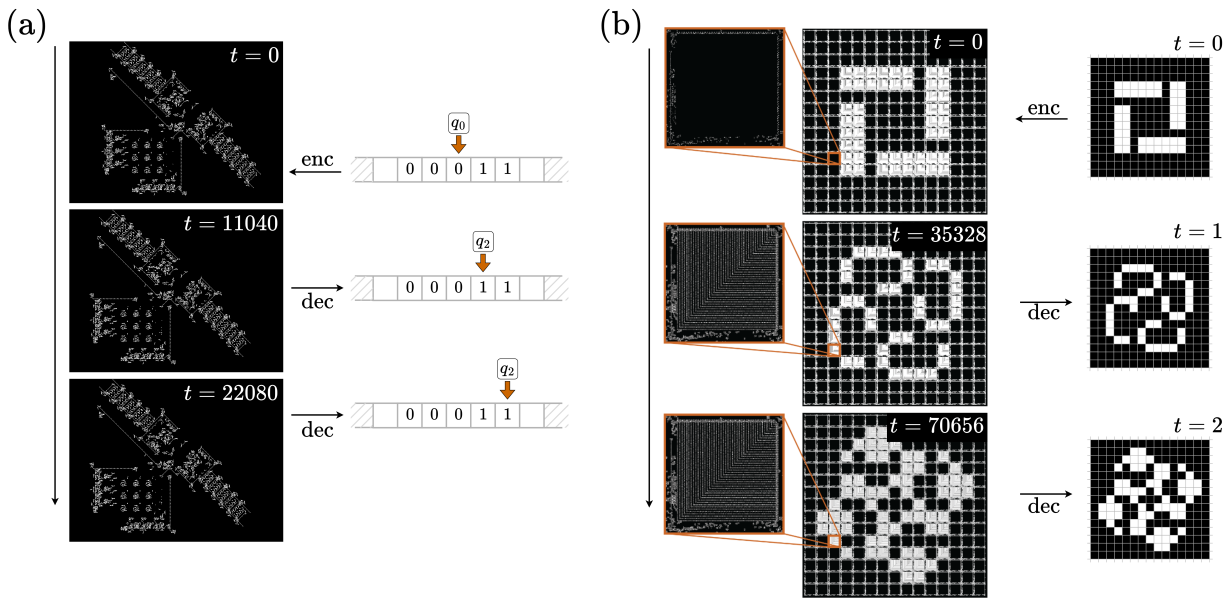


Figure 2: (a) Game of Life (on the left) simulating a Turing machine with 3 states and 3 symbols. (b) Game of Life (on the left) simulating a totalistic CA with two states.

A cellular automaton is called *locally universal* (also Turing universal/Turing complete) if it can locally simulate a universal Turing machine. A $d$-dimensional CA is *globally universal* (also intrinsically universal) if it can globally simulate any other CA in the same dimension. As will become apparent in the next chapter, global universality is a strictly stronger property than local universality.

Whereas local universality is the traditional way one establishes CA's computational capacity, and underlies the proof of Game of Life's or Rule 110's Turing completeness, it is more difficult to define formally. After all, we are comparing two systems with different architectures: unlike Turing machines, cellular automata operate on infinite grids, and have no pre-defined halting state.

Therefore, we start by introducing the notion of global simulation, which captures a different and rather natural idea: that one cellular automaton is able to emulate the entire dynamics of another cellular automaton. In the next section, we aim to develop some basic intuition behind global simulation.

### 6.1 Global Simulation: First Approximation and Examples

We begin with several canonical examples in which one cellular automaton simulates another. These examples will help us build intuition for which transformations of the global map $G$ should be considered natural in the definition of global simulation.

The following example is adapted from [2] and was originally proved in [5]. Before we proceed, we briefly introduce one-way cellular automata.

A 1D CA is *one-way*, if its local function depends only on the central cell and its right neighbour.

**Example 6.1** (Shifts and Time Delays)**.** *For each 1D cellular automaton $\mathcal{A}$ with radius $r = 1$ there exists a one-way CA $\mathcal{B}$ that can simulate it.*

*Proof.* Let $\mathcal{A} = (S^{\mathbb{Z}}, F)$ be a CA with radius 1. We construct a one-way CA $\mathcal{B} = (T^{\mathbb{Z}}, G)$ as follows: $\mathcal{B}$ will simulate one iteration of $\mathcal{A}$ in two steps, first to aggregate information, second to process it. Suppose that $\mathcal{A}$ is given by the local rule $f : S^3 \to S$. We define the states of $\mathcal{B}$ as $S \cup (S \times S)$ and we define its local rule $g$ as follows:

$$\begin{aligned} g(s_1, s_2) &= s_1 s_2 && \text{if } s_1, s_2 \in S \\ &= f(a, b, c) && \text{if } s_1 = ab,\ s_2 = bc \end{aligned}$$

and is otherwise defined arbitrarily.

The simulation is illustrated in Figure 3.

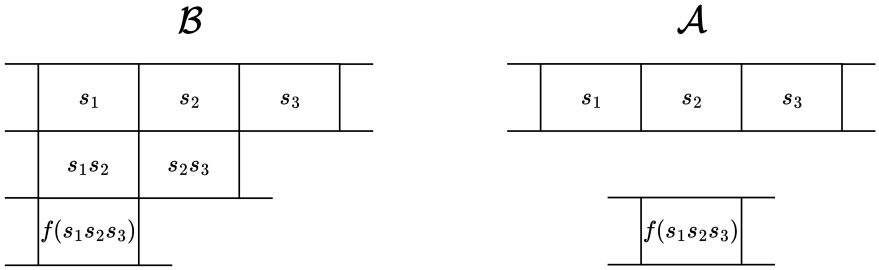


Figure 3: One-way CA simulating a CA with radius $r = 1$.

Notice that in general, the dynamics of $\mathcal{B}$ can be richer, but if the initial configuration $c$ of $\mathcal{B}$ contains only states from $S$; i.e., $c \in S^{\mathbb{Z}} \subseteq (S \cup (S \times S))^{\mathbb{Z}}$ then the dynamics of $\mathcal{B}$ mimics the dynamics of $\mathcal{A}$. Specifically, in order to transform any space-time diagram of $\mathcal{B}$ starting from some $c \in S^{\mathbb{Z}}$ we have to skip every second time-step since $\mathcal{B}$ takes longer to aggregate information due to its smaller neighbourhood. Lastly, in order for the exact cell positions to match, we have to shift

the $k$-th row that we did not skip by $k$ cells to the right. We can summarize the relationship of the two CAs by relating their global rules:

$$\tilde{G}(c) \coloneqq \sigma_{-1} \circ G^2(c) = F(c) \quad \text{for all } c \in S^{\mathbb{Z}}.$$

□

In the example above, in order for $\mathcal{B}$ to simulate $\mathcal{A}$ it needed a richer state space. This raises a natural question:

$$\text{Is it possible for } \mathcal{B} = (T^{\mathbb{Z}^d}, G) \text{ to simulate } \mathcal{A} = (S^{\mathbb{Z}^d}, F), \text{ if } |T| < |S|?$$

This question is crucial in particular in the context of *global universality* as that requires a CA being able to globally simulate automata with arbitrarily large state spaces. To this end, we introduce the "packing operator" which allows us to view a fixed automaton $\mathcal{B}$ as a CA with a richer state space. We have already encountered this method in Example **??**. Below, we add one more example in the explicit context of CA global simulation.

**Example 6.2** (Packing Operator)**.** *Let us consider the "traffic" elementary CA 184 with global map $G$, and the elementary CA 128 with global map $F$ whose local rule $f$ is defined as $f(a, b, c) = abc$. Both CAs are illustrated in Figure 4.*

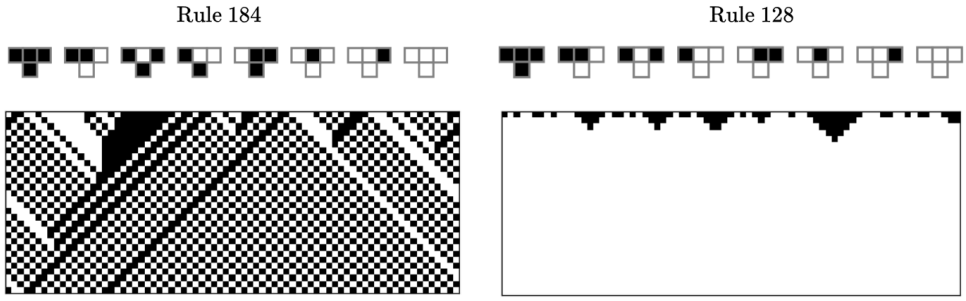


Figure 4: Local rule and a space-time diagram of Rule 184 (Left) and Rule 128 (Right).

*We define a local partial map $\delta : \{0,1\}^2 \rightharpoonup \{0,1\}$ as follows: $\delta(10) = 0$, $\delta(01) = 1$, and $\delta$ is undefined elsewhere. This induces a surjective global partial map $\mathcal{D} : \{0,1\}^{\mathbb{Z}} \rightharpoonup \{0,1\}^{\mathbb{Z}}$, which simply "partitions" each configuration into blocks of two consecutive cells, and uses the local map $\delta$ to decode each of them. Clearly, the domain of $\mathcal{D}$ corresponds to configurations whose supercells only contain the state 01 or 10.*

*Let us see how Rule 184 locally acts on such "supercells" of size 2 that only contain the state 10 or the state 01, after two iterations; we show this in Figure 5.*

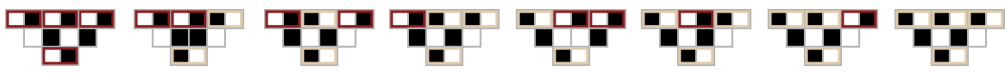

Figure 5: Two iterations of the unravelled local rule of ECA 184 acting on supercells with either the state 10 or 01.

*We can see that indeed on this space Rule 184 mimics the rule table of Rule 128; namely:*

1. *If we start with supercells whose states are in the domain of $\delta$, the outcome is also a supercell whose state in the domain of $\delta$.*

2. *The only time the "phase of the car" 01 is preserved by the central cell, is if it surrounded by two other "cars in the same phase" 01.*

*Therefore, it holds that for every configuration* $c \in \mathbf{2}^{\mathbb{Z}}$*:*

$$\mathcal{D} \circ G^2 \circ \mathcal{D}^{-1}(c) = F(c)$$

*and that this already implies*

$$\mathcal{D} \circ G^{2n} \circ \mathcal{D}^{-1}(c) = F^n(c) \quad \text{for each } n \in \mathbb{N} \text{ and } c \in \mathbf{2}^{\mathbb{Z}}.$$

*We illustrate this relationship in Figure 6.*

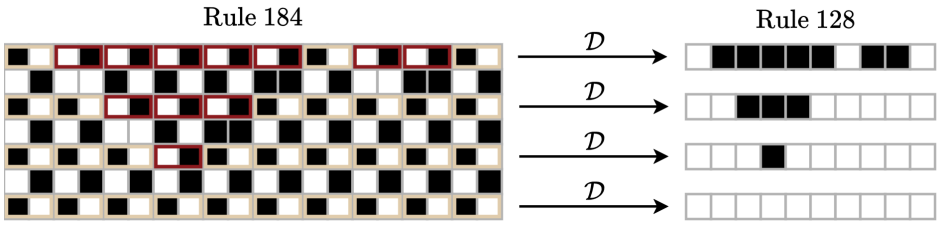


Figure 6: Rule 184 globally simulating Rule 128.

*As a special case, if we define* $\mathcal{E} : \{0,1\}^{\mathbb{Z}} \to \{0,1\}^{\mathbb{Z}}$ *to be a right inverse of* $\mathcal{D}$*, we obtain:*

$$\mathcal{D} \circ G^{2n} \circ \mathcal{E}(c) = F^n(c) \quad \text{for each } n \in \mathbb{N} \text{ and } c \in \mathbf{2}^{\mathbb{Z}}.$$

*Therefore, it seems natural to say that Rule* 184 *globally simulates Rule* 128*.*

To summarize, there are three simple geometric transformations that repeatedly arise when comparing space-time diagrams of cellular automata:

1. A *time-delay*, meaning that one step of $\mathcal{A}$ corresponds to several steps of $\mathcal{B}$,
2. A *shift*, meaning that the relevant information in the diagram of $\mathcal{B}$ drifts through space as time evolves, and
3. A *packing*, meaning that several neighbouring cells of $\mathcal{B}$ should be viewed as one larger "supercell".

These three transformations are not chosen arbitrarily: [3] shows that every geometrical transformation of CA space-time diagrams that preserves temporal causality is a composition of these three operators. We now define the packing operator formally.

## 6.2 Packing Operators

Packings are a family of geometrical transformations of cellular automaton configurations, introduced in full generality in [3], which play a central role in notions of global simulation. Intuitively, if a finite tile $D \subset \mathbb{Z}^d$ induces a tiling of $\mathbb{Z}^d$, then the associated packing operator groups together the cells in each tile into a larger "supercell", thus mapping configurations from $S^{\mathbb{Z}^d}$ to $(S^D)^{\mathbb{Z}^d}$.

**Definition 6.3** (Tilings, Packed Configuration Spaces, and Packing Operators)**.** *Let $D \subset \mathbb{Z}^d$ be a finite set, in this context called a* tile*, and let $V = (\mathbf{v}_1, \dots, \mathbf{v}_d)$ be a sequence of linearly independent vectors in $\mathbb{Z}^d$. We write*

$$M_V = (\mathbf{v}_1 \mid \mathbf{v}_2 \mid \cdots \mid \mathbf{v}_d) \in \mathbb{Z}^{d \times d}$$

*for the matrix whose columns are the vectors of $V$. We say that $(D, V)$ is a* tiling *of $\mathbb{Z}^d$ if the family of translates*

$$\{D + M_V \mathbf{z} \mid \mathbf{z} \in \mathbb{Z}^d\}$$

*forms a partition of $\mathbb{Z}^d$, see Figure 7.*

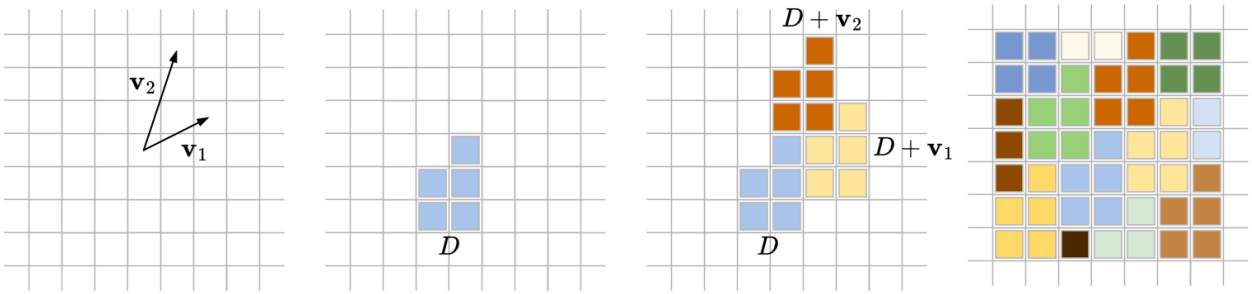


Figure 7: (left) Vectors $\mathbf{v}_1 = (2, 1)$, $\mathbf{v}_2 = (1, 3)$ spanning a sublattice in $\mathbb{Z}^2$. (middle-left) The tile $D = \{(0, 0), (1, 0), (0, 1), (1, 1), (1, 2)\}$. (middle-right) Translation of $D$ by $\mathbf{v}_1$ and by $\mathbf{v}_2$. (right) Tiling of $\mathbb{Z}^2$ induced by $(D, (\mathbf{v}_1, \mathbf{v}_2))$.

*Let $S$ be a finite set of states. We say that $(S^D)^{\mathbb{Z}^d}$ is the* packed configuration space *associated with the tiling $(D, V)$. Here, an element of $S^D$ represents the state of one supercell. Thus, a configuration $c \in (S^D)^{\mathbb{Z}^d}$ assigns to each packed coordinate $\mathbf{z} \in \mathbb{Z}^d$ a pattern $c(\mathbf{z}) : D \to S$. There is a natural way to relate the geometry of the packed configuration space $(S^D)^{\mathbb{Z}^d}$ to the one of $S^{\mathbb{Z}^d}$ via an isomorphism called the* packing operator

$$o_{\langle D,V \rangle} : S^{\mathbb{Z}^d} \to (S^D)^{\mathbb{Z}^d}$$

*defined by*

$$o_{\langle D,V \rangle}(c)(\mathbf{z})(\mathbf{p}) = c(\mathbf{p} + M_V \, \mathbf{z})$$

*for every configuration $c \in S^{\mathbb{Z}^d}$, every packed coordinate $\mathbf{z} \in \mathbb{Z}^d$, and every $\mathbf{p} \in D$. We sometimes write $(S_V^D)^{\mathbb{Z}^d}$ instead of $(S^D)^{\mathbb{Z}^d}$ if we want to highlight the underlying geometry of the packed space with respect to the original space $S^{\mathbb{Z}^d}$. The packed configuration space is illustrated in Figure 8.*

The packing operator is a bijection. Indeed, since the translates $D + M_V \mathbf{z}$ partition $\mathbb{Z}^d$, each site $\mathbf{u} \in \mathbb{Z}^d$ can be written uniquely as $\mathbf{u} = \mathbf{p} + M_V \mathbf{z}$ with $\mathbf{p} \in D$ and $\mathbf{z} \in \mathbb{Z}^d$. Moreover, it is not difficult to check it is continuous.

**Observation 6.4** (Packing operator and shifts)**.** *Let $(D, V)$ be a tiling of $\mathbb{Z}^d$ with $V = (\mathbf{v}_1, \dots, \mathbf{v}_d)$, and let $S$ be a finite set. For each $\mathbf{v} \in \mathbb{Z}^d$ we denote by $\tilde{\sigma}_{\mathbf{v}} : (S^D)^{\mathbb{Z}^d} \to (S^D)^{\mathbb{Z}^d}$ the shift map on the packed configuration space and by $\sigma_{\mathbf{v}} : S^{\mathbb{Z}^d} \to S^{\mathbb{Z}^d}$ the shift map on the original space. Then,*

$$\tilde{\sigma}_{\mathbf{v}} \circ o_{\langle D,V \rangle} = o_{\langle D,V \rangle} \circ \sigma_{M_V \, \mathbf{v}}.$$

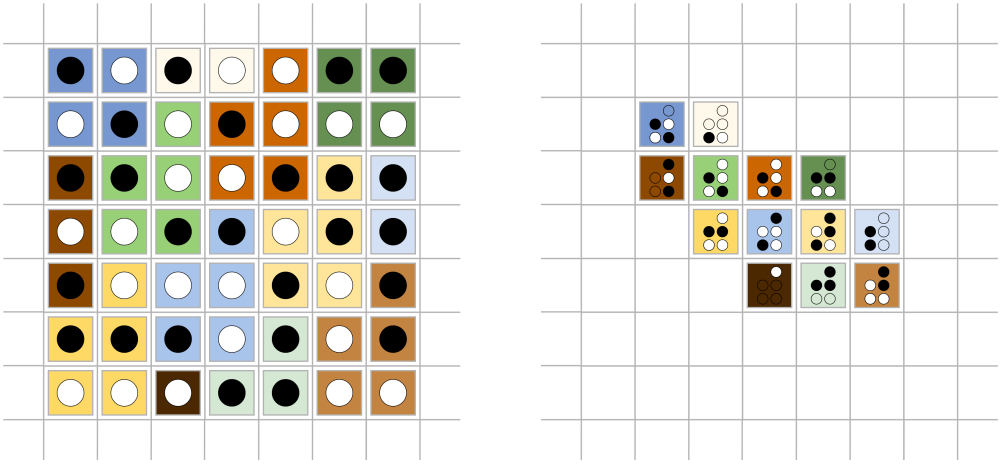

Figure 8: The tiling $(D, V)$ from Figure 7 induces a packing operator $o_{\langle D,V\rangle} : \mathbf{2}^{\mathbb{Z}^2} \to (\mathbf{2}_V^D)^{\mathbb{Z}^2}$. (left) Configuration $c \in \mathbf{2}^{\mathbb{Z}^2}$. White and black circles illustrate the states 0 and 1; we keep the background colouring to show the space tiling. (right) The outcome $o_{\langle D,V\rangle}(c)$; the states are aggregated into supercells according to the tile $D$.

It is clear that the packing operator is not a strong conjugacy as it does not necessarily commute with all shifts on the original space. However, it does have the property that if a CA global rule $G$ is transformed by the packing operator into $o_{\langle D,V\rangle} \circ G \circ o^{-1}_{\langle D,V\rangle}$ the resulting map is again a global map of a CA on this packed space, as can easily be verified using Observation 6.4.

**Definition 6.5** (Packed Cellular Automaton)**.** *Let $\mathcal{B} = (T^{\mathbb{Z}^d}, G)$ be a cellular automaton and $(D, V)$ is a tiling of $\mathbb{Z}^d$. We define the packed cellular automaton*

$$\mathcal{B}^{\langle D,V\rangle} = ((T^D)^{\mathbb{Z}^d}, G^{\langle D,V\rangle})$$

*where*

$$G^{\langle D,V\rangle} = o_{\langle D,V\rangle} \circ G \circ o^{-1}_{\langle D,V\rangle}.$$

Clearly, $\mathcal{B}$ and $\mathcal{B}^{\langle D,V\rangle}$ are conjugate via the packing operator $o_{\langle D,V\rangle}$. It is straightforward to verify that $G^{\langle D,V\rangle}$ is a continuous map which commutes with all the shifts on the packed space, and therefore, that $\mathcal{B}^{\langle D,V\rangle}$ is indeed a cellular automaton.

More generally, when we study the global simulation capacity of a cellular automaton $\mathcal{B}$ we further allow a time-delay $\tau \in \mathbb{N}^+$ and a spatial shift $\mathbf{v} \in \mathbb{Z}^d$, to transform the global rule $G$ into

$$G^{\langle D,V\rangle,\mathbf{v},\tau} = o_{\langle D,V\rangle} \circ \sigma_{\mathbf{v}} \circ G^{\tau} \circ o^{-1}_{\langle D,V\rangle}.$$

Thus, packing, time-delay, and shifting transform a CA into another CA acting on a packed configuration space.

### 6.3 Computationally Admissible Transformations

We now turn our attention to specifying the encoder and decoder maps which witness that a CA $\mathcal{B}$ globally simulates $\mathcal{A}$, and which are "computationally reasonable". For that, it is convenient to use the following notion.

**Definition 6.6** (Cellular transformaton)**.** *Let $S$ and $T$ be finite sets. A mapping $\Phi : S^{\mathbb{Z}^d} \to T^{\mathbb{Z}^d}$ is called a* cellular transformaton *if there exist a neighbourhood $N = (\mathbf{n}_1, \dots, \mathbf{n}_k) \subseteq (\mathbb{Z}^d)^k$ and a local rule $\phi : S^k \to T$ such that for every $c \in S^{\mathbb{Z}^d}$ and every $\mathbf{v} \in \mathbb{Z}^d$,*

$$\Phi(c)(\mathbf{v}) = \phi\big(c(\mathbf{v} + \mathbf{n}_1), \dots, c(\mathbf{v} + \mathbf{n}_k)\big).$$

This is simply the natural generalization of a cellular automaton when the input and output state spaces need not coincide. As in the Curtis–Hedlund–Lyndon theorem, one shows that cellular transformata are exactly the continuous maps that commute with all shifts. Likewise, after applying suitable packing operators, the encoders and decoders that appear in global simulation may be viewed as cellular transformata. We summarize this in the following "packed" version of the Curtis-Hedlund-Lyndon Theorem.

**Theorem 6.7** (Packing version of the Curtis–Hedlund–Lyndon Theorem for cellular transformata)**.** *Let $S, T$ be finite sets and $V = (\mathbf{v}_1, \dots, \mathbf{v}_d)$, $W = (\mathbf{w}_1, \dots, \mathbf{w}_d)$ two sequences of $d$ linearly independent vectors in $\mathbb{Z}^d$; we denote their corresponding matrices by $M_V$ and $M_W$. Let $(D, V)$ and $(E, W)$ be two tilings of $\mathbb{Z}^d$ and let $\Phi : S^{\mathbb{Z}^d} \to T^{\mathbb{Z}^d}$ be an arbitrary map. Then, the following two conditions are equivalent:*

*(i) $\Phi$ is continuous and for each $\mathbf{v} \in \mathbb{Z}^d$ satisfies $\Phi \circ \sigma_{M_V \mathbf{v}} = \sigma_{M_W \mathbf{v}} \circ \Phi$.*

*(ii) $\Phi^{\langle D,V\rangle}_{\langle E,W\rangle} \coloneqq o_{\langle E,W\rangle} \circ \Phi \circ \big(o_{\langle D,V\rangle}\big)^{-1} : (S^D)^{\mathbb{Z}^d} \to (T^E)^{\mathbb{Z}^d}$ is a cellular transformaton.*

**Exercise 6.8.** *Prove the Packing Version of the Curtis–Hedlund–Lyndon Theorem for Cellular Transformata.*

In particular, such maps are computationally reasonable in exactly the sense one would expect: they act locally and shift-equivariantly on packed configuration spaces.

**Definition 6.9** (Computationally admissible map)**.** *We call a map $\Phi : S^{\mathbb{Z}^d} \to T^{\mathbb{Z}^d}$ a* computationally admissible map *if it satisfies the equivalent conditions of Theorem 6.7. That is, if it can be viewed as a cellular transformaton between some packed versions of the spaces $S^{\mathbb{Z}^d}$ and $T^{\mathbb{Z}^d}$.*

Finally, it will be useful to generalize the notion of cellular transformata also to partial maps.

**Definition 6.10** (Partial cellular transformaton)**.** *Let $S, T$ be finite sets and let $X \subseteq S^{\mathbb{Z}^d}$. A map $\Phi : X \to T^{\mathbb{Z}^d}$ is called a* partial cellular transformaton *if there exist a neighbourhood $N = (\mathbf{n}_1, \dots, \mathbf{n}_k) \subseteq (\mathbb{Z}^d)^k$ and a local rule $\phi : S^k \to T$ such that for every $c \in X$ and every $\mathbf{v} \in \mathbb{Z}^d$,*

$$\Phi(c)(\mathbf{v}) = \phi\big(c(\mathbf{v} + \mathbf{n}_1), \dots, c(\mathbf{v} + \mathbf{n}_k)\big).$$

*In the applications below, $X$ will always be a subshift of finite type.*

In the next section, we will finally introduce the formal definition of global simulation, where we require the witnessing encoder and decoder to be exactly cellular transformata on some packed spaces. Before we proceed, we show an example of "what can go wrong" when the encoder or decoder is a computationally unrestricted map.

**Example 6.11** (Computationally unrestricted decoder)**.** *Let $\mathcal{A} = (\mathbf{2}^{\mathbb{Z}}, F)$ be an arbitrary 1D 2-state cellular automaton. Consider also the left shift CA $\mathcal{B} = (\mathbf{2}^{\mathbb{Z}}, \sigma)$. We will describe that with an unrestricted decoder it may seem as though $\mathcal{B}$ globally simulates $\mathcal{A}$. Let*

$$c = \dots x_{-2}\, x_{-1}.x_0\, x_1 \dots \in \{0,1\}^{\mathbb{Z}}$$

*be a configuration of $\mathcal{A}$ (in this notation, the position after the dot . corresponds to 0). Define the encoder $\mathcal{E} : \mathbf{2}^{\mathbb{Z}} \to \mathbf{2}^{\mathbb{Z}}$ by*

$$\mathcal{E}(c) = \ldots 0\, x_{-2}\, 0\, x_{-1}.1\, x_0\, 0\, x_1\, 0 \ldots$$

*That is, $\mathcal{E}(c)$ interleaves the symbols of $c$ with 0's, except that at the origin we place a distinguished marker 1.*

*Now apply $k$ steps of the shift map $G$. Since $G$ is just the left shift, the marker 1 moves by $k$ positions. One may then define a decoder $\mathcal{D} : \mathbf{2}^{\mathbb{Z}} \to \mathbf{2}^{\mathbb{Z}}$ so that, whenever the marker is found in the position corresponding to time $k$, the decoder outputs*

$$\mathcal{D}(G^k(\mathcal{E}(c))) = F^k(c).$$

*In this way, it looks as though the simple shift CA $\mathcal{B}$ simulates the dynamics of the arbitrary CA $\mathcal{A}$. However, this is entirely misleading as all the genuine computational content is hidden inside the decoder, which reads the position of the marker, reconstructs the integer $k$, and then computes $F^k(c)$ by itself. Thus, the apparent "simulation" carries no meaningful information about the computational power of $\mathcal{B}$.*

*This example illustrates why, in any robust definition of global simulation, the encoder and decoder must themselves be computationally restricted.*

### 6.4 Global Simulation Definition and Basic Properties

We can now state the formal definition.

**Definition 6.12** (Global simulation of cellular automata)**.** *Let $\mathcal{A} = (S^{\mathbb{Z}^d}, F)$ and $\mathcal{B} = (T^{\mathbb{Z}^d}, G)$ be $d$-dimensional cellular automata. We say that $\mathcal{B}$* globally simulates *$\mathcal{A}$ if there exist a vector $\mathbf{v} \in \mathbb{Z}^d$, a time-delay $\tau \in \mathbb{N}^+$, and a partial surjective map $\mathcal{D} : T^{\mathbb{Z}^d} \rightharpoonup S^{\mathbb{Z}^d}$ such that for all $c \in \mathrm{dom}\,(\mathcal{D})$*

$$\mathcal{D} \circ \sigma_{\mathbf{v}} \circ G^{\tau}(c) = F \circ \mathcal{D}(c).$$

*Further, we require $\mathcal{D}$ to be a computationally reasonable map in the following sense:*

1. *There exists a tiling $(D, V)$ of $\mathbb{Z}^d$ such that $\tilde{\mathcal{D}} = \mathcal{D} \circ o^{-1}_{\langle D,V\rangle} : (T^D)^{\mathbb{Z}^d} \rightharpoonup S^{\mathbb{Z}^d}$ is a partial cellular transformaton between the packed space of $(T^D_V)^{\mathbb{Z}^d}$ and $S^{\mathbb{Z}^d}$ whose domain $\mathrm{dom}\,(\tilde{\mathcal{D}}) \subseteq (T^D)^{\mathbb{Z}^d}$ is a subshift of finite type on the packed space,*
2. *$\mathcal{D}$ has a right inverse $\mathcal{E} : S^{\mathbb{Z}^d} \to T^{\mathbb{Z}^d}$; i.e., $\mathcal{D} \circ \mathcal{E}$ is an identity on $S^{\mathbb{Z}^d}$ and $\mathcal{E}$ is a computationally admissible map.*

*We write $\mathcal{A} \preccurlyeq \mathcal{B}$ and we call $\mathcal{D}$ the decoder and $\mathcal{E}$ the encoder which witness the global simulation.*

We can notice that $\mathcal{E}$ in the previous definition can also be viewed as a cellular transformaton between some packing of space $S^{\mathbb{Z}^d}$ and some packing of $T^{\mathbb{Z}^d}$.

**Definition 6.13** (Globally universal cellular automaton)**.** *A $d$-dimensional cellular automaton $\mathcal{B}$ is* globally universal *if $\mathcal{A} \preccurlyeq \mathcal{B}$ for every $d$-dimensional cellular automaton $\mathcal{A}$.*

The following proposition shows that whenever $\mathcal{B}$ globally simulates $\mathcal{A}$ we obtain an analogous relationship for arbitrary iterations of their global maps, which indeed yields the initial picture we illustrated in Figure 1.

**Proposition 6.14.** *If $\mathcal{A} \preccurlyeq \mathcal{B}$, then for every $n \in \mathbb{N}$, every configuration $x \in \mathrm{dom}\,(\mathcal{D})$ one has*

$$\mathcal{D} \circ (\sigma_{\mathbf{v}} \circ G^{\tau})^n(x) = F^n \mathcal{D}(x).$$

*Proof.* Let $c \in S^{\mathbb{Z}^d}$ and $x \in \mathcal{D}^{-1}(c)$. We argue by induction on $n$. The case $n = 0$ is immediate, since $\mathcal{D}(x) = c = F^0(c)$.

Now assume the claim holds for some $n \in \mathbb{N}$, and set $y = (\sigma_{\mathbf{v}} \circ G^\tau)^n(x)$. Then $\mathcal{D}(y) = F^n(c)$ by the induction hypothesis, so $y \in \mathcal{D}^{-1}(F^n(c))$. Applying the defining property of global simulation to the configuration $F^n(c)$ and the point $y$, we obtain

$$\mathcal{D}\big(\sigma_{\mathbf{v}}(G^\tau(y))\big) = F(F^n(c)) = F^{n+1}(c).$$

Since $\sigma_{\mathbf{v}}(G^\tau(y)) = (\sigma_{\mathbf{v}} \circ G^\tau)^{n+1}(x)$, the induction closes. □

As we discuss in the following remark, the relation $\preccurlyeq$ behaves as one would hope.

**Remark 6.15.** *Reflexivity of $\preccurlyeq$ is immediate: take $\mathbf{v} = 0$, $\tau = 1$, and let both encoder and decoder be the identity.*

*Transitivity is standard once one rewrites Definition 6.12 in the usual packed block-code formalism of intrinsic simulation. Since we will not need the full transitivity argument below, we omit it here.*

We finish this section by developing further intuition for the definition of global simulation through the algebraic perspective of subsystems and quotients.

## 6.5 Algebraic View of Global CA Simulation

We finish this section by interpreting global simulation algebraically in the language of geometric transformations, subsystems, and factors. This shows that global simulation is not unrelated to the dynamical comparison tools studied earlier; rather, it is obtained by combining them in a computationally constrained way.

Suppose for the rest of this section that the CA $\mathcal{B} = (T^{\mathbb{Z}^d}, G)$ globally simulates the CA $\mathcal{A} = (S^{\mathbb{Z}^d}, F)$ via a shift vector $\mathbf{v} \in \mathbb{Z}^d$, a time-delay $\tau \in \mathbb{Z}_{\geq 1}$, and a decoder $\mathcal{D} : T^{\mathbb{Z}^d} \dashrightarrow S^{\mathbb{Z}^d}$ satisfying the conditions of Definition 6.12.

We can decompose the decoder as $\mathcal{D} = \tilde{\mathcal{D}} \circ o_{\langle D,V\rangle}$ where $(D, V)$ is some tiling of $\mathbb{Z}^d$ and $\tilde{\mathcal{D}} : (T^D)^{\mathbb{Z}^d} \dashrightarrow S^{\mathbb{Z}^d}$ is a partial cellular transformaton whose domain $\mathrm{dom}\,(\tilde{\mathcal{D}}) \subseteq (T^D)^{\mathbb{Z}^d}$ is a subshift of finite type. Then, the defining condition of global simulation may be rewritten as

$$\tilde{\mathcal{D}} \circ o_{\langle D,V\rangle} \circ \sigma_{\mathbf{v}} \circ G^\tau \circ o^{-1}_{\langle D,V\rangle} = F \circ \tilde{\mathcal{D}}.$$

We can notice that $\tilde{G} = o_{\langle D,V\rangle} \circ \sigma_{\mathbf{v}} \circ G^\tau \circ o^{-1}_{\langle D,V\rangle}$ is a global map of a CA $\mathcal{B}^{\langle D,V\rangle,\mathbf{v},\tau}$ which is a transformation of $\mathcal{B}$ by a shift, time-delay and a packing; and that now, $\tilde{\mathcal{D}}$ is simply a partially defined cellular transformaton from the configuration space of $\mathcal{B}^{\langle D,V\rangle,\mathbf{v},\tau}$ onto the configuration space of $\mathcal{A}$. Equivalently, we have that the following diagram commutes:

$$\begin{array}{ccc} \mathrm{dom}\,(\tilde{\mathcal{D}}) & \xrightarrow{\tilde{G}} & \mathrm{dom}\,(\tilde{\mathcal{D}}) \\ {\scriptstyle\tilde{\mathcal{D}}}\big\downarrow & & \big\downarrow{\scriptstyle\tilde{\mathcal{D}}} \\ S^{\mathbb{Z}^d} & \xrightarrow[F]{} & S^{\mathbb{Z}^d} \end{array}$$

This immediately yields the following proposition.

**Observation 6.16.** *Suppose $\mathcal{A} \preccurlyeq \mathcal{B}$, and let $\tilde{G} = o_{\langle D,V\rangle} \circ \sigma_{\mathbf{v}} \circ G^\tau \circ o^{-1}_{\langle D,V\rangle}$ and $\tilde{\mathcal{D}} = \mathcal{D} \circ o^{-1}_{\langle D,V\rangle}$ be as above. Then:*

1. $\mathrm{dom}\,(\tilde{\mathcal{D}})$ *is invariant under* $\widetilde{G}$. *In particular,*

$$\left(\mathrm{dom}\,(\tilde{\mathcal{D}}), \widetilde{G}\Big|_{\mathrm{dom}\,(\tilde{\mathcal{D}})}\right)$$

*is a subsystem of the cellular automaton* $\mathcal{B}^{\langle D,V\rangle,\mathbf{v},\tau} = ((T^D)^{\mathbb{Z}^d}, \widetilde{G})$.

2. *The decoder* $\tilde{\mathcal{D}} : \mathrm{dom}\,(\tilde{\mathcal{D}}) \to S^{\mathbb{Z}^d}$ *is a continuous surjection satisfying* $\tilde{\mathcal{D}} \circ \widetilde{G}\Big|_{\mathrm{dom}\,(\tilde{\mathcal{D}})} = F \circ \tilde{\mathcal{D}}$. *Hence* $\mathcal{A}$ *is a factor of the dynamical system*

$$\left(\mathrm{dom}\,(\tilde{\mathcal{D}}), \widetilde{G}\Big|_{\mathrm{dom}\,(\tilde{\mathcal{D}})}\right).$$

**Exercise 6.17** (Algebraic view of global simulation)**.** *Prove the proposition above.*

Thus, whenever $\mathcal{B}$ globally simulates $\mathcal{A}$, the system $\mathcal{A}$ appears as a factor of a suitable subsystem of a geometrically transformed version of $\mathcal{B}$. In this sense, global simulation is built from the same comparison principles that we studied earlier, but with additional computational restrictions imposed on the corresponding witnessing maps. This viewpoint motivates the following operators on families of cellular automata.

**Definition 6.18** (Geometric transformation, subsystem, and factor operators)**.** *Let* $\mathcal{F}$ *be a family of* $d$*-dimensional cellular automata. We define:*

$$\mathcal{G}(\mathcal{F}) = \left\{\mathcal{A} \;\middle|\; \mathcal{A} = \mathcal{B}^{\langle D,V\rangle,\mathbf{v},\tau} \text{ for some } \mathcal{B} \in \mathcal{F},\ \mathbf{v} \in \mathbb{Z}^d,\ \tau \in \mathbb{N}^+ \text{ and some tiling } (D,V) \text{ of } \mathbb{Z}^d\right\},$$
$$\mathcal{S}(\mathcal{F}) = \{\mathcal{A} \mid \mathcal{A} \text{ embeds into } \mathcal{B} \text{ for some } \mathcal{B} \in \mathcal{F}\},$$
$$\mathcal{H}(\mathcal{F}) = \{\mathcal{A} \mid \mathcal{A} \text{ is a factor of } \mathcal{B} \text{ for some } \mathcal{B} \in \mathcal{F}\}.$$

To capture global simulation rather than arbitrary dynamical comparison, we must restrict the subsystem and factor operations to computationally feasible ones.

**Definition 6.19** (Computationally feasible subsystem and factor operators)**.** *We define* $\mathcal{S}^{\mathrm{comp}}(\mathcal{F})$ *to consist of those systems obtained from elements of* $\mathcal{F}$ *by passing to subsystems given by subshifts of finite type.*

*We define* $\mathcal{H}^{\mathrm{comp}}(\mathcal{F})$ *to consist of those factors obtained by factor maps* $\tilde{\mathcal{D}}$ *which are cellular transformata and admit a right inverse* $\mathcal{E}$ *which, again after suitable packing transformations, is also a cellular transformaton.*

With this notation, Proposition 6.16 may be summarized informally as follows:

$$\mathcal{B} \text{ globally simulates } \mathcal{A} \quad \text{if and only if} \quad \mathcal{A} \in \mathcal{H}^{\mathrm{comp}}\mathcal{S}^{\mathrm{comp}}\mathcal{G}(\{\mathcal{B}\}).$$

To summarize, global simulation is obtained by first allowing simple geometric transformations of $\mathcal{B}$, then passing to a computationally feasible subsystem, and finally taking a computationally feasible factor.

This formulation is useful for two reasons. First, it clarifies the structure of the notion itself: global simulation is not a completely new comparison relation, but a computationally constrained composition of familiar ones. Second, it suggests a route to generalization as the notions of subsystems and factors are natural for any family of dynamical systems. Thus, for other classes of dynamical systems, one may attempt to define suitable geometric transformations together with admissible subsystem and factor operations, and thereby obtain an analogous notion of global simulation.

## 6.6 Dynamical and Computational Classes: Global Perspective

The purpose of the computational dynamical systems formalism goes beyond establishing whether a given dynamical system is or is not universal. The broader aim is to relate *dynamical classes* to *computational classes*. In the setting of global simulation, this means asking questions of the following form: if a dynamical system belongs to a natural dynamical class, such as measure-preserving systems, systems with zero topological entropy, or various forms of mixing systems, what kinds of systems can it simulate via encoders and decoders of controlled complexity? The hope is that one can eventually attach to each dynamical class a corresponding bound on computational power, so that statements of the form

"systems of this dynamical type can realize at most this computational class"

can be rendered into precise theorems.

The formalism of encoders, decoders, and slowdown functions is precisely what makes such questions mathematically meaningful. Without explicit complexity bounds on the translation layer, one can conceal arbitrary computation inside the encoder or decoder and thereby trivialize the question of what the dynamics itself is doing. Once the translation layer is constrained, however, the dynamical properties of the system begin to matter in a genuine way. In the setting of smooth dynamical systems this has already led to non-universality results for broad classes of systems, as well as to explicit time-complexity bounds in certain low-dimensional situations.

In the context of global simulation of cellular automata, we now show an example of establishing exactly such a relationship between a dynamical property, namely reversibility, and the inability to simulate any irreversible CA. This result was first established by Hertling [4].

**Proposition 6.20.** *No reversible cellular automaton is globally universal.*

*Proof.* Let $\mathcal{O} = (\{0,1\}^{\mathbb{Z}^d}, O)$ be the constant cellular automaton defined by $O(c) = 0^{\mathbb{Z}^d}$ for every configuration $c \in \{0,1\}^{\mathbb{Z}^d}$. This cellular automaton is clearly irreversible.

Suppose for contradiction that a reversible cellular automaton $\mathcal{R} = (S^{\mathbb{Z}^d}, G)$ globally simulates $\mathcal{O}$ via an encoder $\mathcal{E}$, a decoder $\mathcal{D}$, a shift vector $\mathbf{v}$, and a time-delay $\tau$. Let $M_1, M_2$ be the full-rank matrices witnessing the shift-compatibility of the encoder.

Call a configuration $c \in \{0,1\}^{\mathbb{Z}^d}$ $M_1$*-periodic* if $\sigma_{M_1\mathbf{w}}(c) = c$ for every $\mathbf{w} \in \mathbb{Z}^d$. Since $\mathcal{E}$ commutes with shifts up to $M_1$ and $M_2$, the image of every $M_1$-periodic configuration under $\mathcal{E}$ is $M_2$-periodic. There are only finitely many $M_2$-periodic configurations, because such a configuration is determined by its values on a fundamental domain of the lattice $M_2\mathbb{Z}^d$.

Now consider the map $H = \sigma_{\mathbf{v}} \circ G^{\tau}$. Since $\mathcal{R}$ is reversible, $G$ is bijective, hence $H$ is bijective. Moreover, $H$ preserves the set of $M_2$-periodic configurations, because both shifts and cellular automata preserve periodicity. Therefore, when restricted to this finite set, $H$ is a permutation. In particular, every $M_2$-periodic configuration is periodic under $H$.

Choose any $M_1$-periodic configuration $c \in \{0,1\}^{\mathbb{Z}^d}$ with $c \neq 0^{\mathbb{Z}^d}$, and let $x = \mathcal{E}(c)$. Then $x$ is $M_2$-periodic, so there exists $m \in \mathbb{N}_{>0}$ such that $H^m(x) = x$. Applying Proposition 6.14, we obtain $\mathcal{D}(H^m(x)) = O^m(c) = 0^{\mathbb{Z}^d}$. But $H^m(x) = x$, so $\mathcal{D}(x) = 0^{\mathbb{Z}^d}$. On the other hand, since $\mathcal{D} \circ \mathcal{E} = \mathrm{id}$, we also have $\mathcal{D}(x) = \mathcal{D}(\mathcal{E}(c)) = c$. Thus $c = 0^{\mathbb{Z}^d}$, a contradiction.

Therefore, no reversible cellular automaton can globally simulate $\mathcal{O}$, and hence no reversible cellular automaton is globally universal. □

## 6.7 Global Universality

The negative result above should not be read as saying that global universality is rare or impossible. On the contrary, globally universal cellular automata exist in every dimension. In dimension one,

there are explicit globally universal cellular automata, one of the earliest ones constructed in the seminal work of Culik and Yu [5]. In dimension $d = 2$, Banks provided a construction of a globally universal CA with only two states [1]. These positive results show that the definition of global simulation is not vacuous: there really are local dynamical laws whose space-time diagrams can reproduce the complete space-time diagrams of arbitrary cellular automata in the same dimension, after suitable packing, shifting, and time rescaling. What Proposition 6.20 shows is instead that reversibility is a genuine dynamical restriction: a reversible CA cannot be globally universal within the same dimension, because it cannot faithfully simulate irreversible dynamics such as the constant CA.

There is, however, an important positive result for reversible cellular automata once one allows an additional spatial dimension. Toffoli [6] showed that every $d$-dimensional CA can be strongly embedded into a reversible CA of dimension $d+1$. The intuition is that the extra dimension can be used to store the information that an irreversible CA would otherwise erase: rather than destroying the past, the reversible simulator records enough of the simulated space-time history to make the global evolution invertible.

## Summary

In this chapter, we introduced global simulation as one way of making precise what it means for a cellular automaton to carry out the computation of another cellular automaton. The central point was that simulation is not merely a vague resemblance between space-time diagrams: it requires an explicit translation layer, consisting of an encoder and a decoder, together with simple geometric transformations such as time rescaling, shifting, and packing. By requiring these translation maps to be computationally admissible, we avoid hiding all the computational work inside the translation layer.

From this perspective, global simulation gives a way to assess the computational capacity of cellular automata. A CA is globally universal if, after suitable admissible transformations, it can reproduce the full dynamics of any other CA. Algebraically, this means that the simulated automaton appears as a factor of a suitable subsystem of a geometrically transformed version of the simulator.

Formalizing the notion of global simulation allows us to prove negative results. In particular, we saw that reversible cellular automata cannot be globally universal, since they cannot globally simulate irreversible dynamics such as the constant cellular automaton. Results of this kind suggest a broader program: to relate dynamical properties of cellular automata, such as reversibility, entropy, or mixing, to their ability to globally simulate other systems.

In the next chapter, we turn to local simulation. Instead of comparing cellular automata with other cellular automata acting on an entire infinite grid, we will compare cellular automata with local models of computation, such as Turing machines and related systems. This will lead us to the more classical notion of Turing universality.

## References


[1] Banks, E. R. Universality in cellular automata. In *Proceedings of the 11th Annual Symposium on Switching and Automata Theory* (Santa Monica, California, 1970), IEEE, pp. 194–215.

[2] Cotler, J., Hongler, C., and Hudcová, B. Self-replication and Computational Universality. *arXiv preprint arXiv:2510.08342* (2025).

[3] Delorme, M., Mazoyer, J., Ollinger, N., and Theyssier, G. Bulking i: an abstract theory of bulking. *Theoretical Computer Science 412*, 30 (2011), 3866–3880.

[4] Hertling, P. Embedding cellular automata into reversible ones. *Unconventional models of computation* (1998), 243.

[5] Jürgen, A., and Culik II, K. A simple universal cellular automaton and its one-way and totalistic version. *Complex Systems 1* (1987), 1–16.

[6] Toffoli, T. Computation and construction universality of reversible cellular automata. *Journal of Computer and System Sciences 15*, 2 (1977), 213–231.