Deterministic computations whose history is independent of the order of asynchronous updating

Peter Gács*
Boston University
gacs@bu.edu

June 12, 2018

Abstract

Consider a network of processors (sites) in which each site x has a finite set $N(x)$ of neighbors. There is a transition function f that for each site x computes the next state $\xi(x)$ from the states in $N(x)$. But these transitions (updates) are applied in arbitrary order, one or many at a time. If the state of site x at time t is $\eta(x, t)$ then let us define the sequence $\zeta(x, 0)$, $\zeta(x, 1)$, ... by taking the sequence $\eta(x, 0)$, $\eta(x, 1)$, ..., and deleting each repetition: each element equal to the preceding one. The function f is said to have *invariant histories* if the sequence $\zeta(x, i)$, (while it lasts, in case it is finite) depends only on the initial configuration, not on the order of updates.

This paper shows that though the invariant history property is typically undecidable, there is a useful simple sufficient condition, called *commutativity*: For any configuration, for any pair x, y of neighbors, if the updating would change both $\xi(x)$ and $\xi(y)$ then the result of updating first x and then y is the same as the result of doing this in the reverse order. This fact is related to the confluence property of term-rewriting systems, and well as sandpile theory in statistical physics.

We will also show a simple simulation of an arbitrary synchronous computation by a commutative asynchronous one.

*Partially supported by NSF grant CCR-920484

1 Introduction

Consider a set $\mathbb{C}$ of *sites* (processors) in which each site x has a set $\mathbb{S}$ of possible *local states*. An arbitrary function $\xi \in \mathbb{S}^{\mathbb{C}}$ is called a *space-configuration* (or simply "configuration", or "global state"). The value $\xi(x)$ is the state of site x in ξ. A *neighborhood function* $N : \mathbb{C} \to 2^{\mathbb{C}}$ assigns to each site x, a set $N(x)$ called its *neighborhood*. A function $f : \mathbb{S}^{\mathbb{C}} \to \mathbb{S}^{\mathbb{C}}$ is called a *transition function* if $f(\xi)(x)$ depends only on $\xi \restriction N(x)$:

$$\xi_1 \restriction N(x) = \xi_2 \restriction N(x) \Rightarrow f(\xi_1)(x) = f(\xi_2)(x).$$

The transition function determines a possible "next" configuration from the "current" one. The 4-tuple

$$\mathbf{A} = (\mathbb{C}, \mathbb{S}, N, f) \tag{1}$$

will be called an *automaton* (not necessarily a finite one). If all sets $N(x)$ are finite then the system is said to have *finite neighborhoods*. This is actually a property of the transition function f itself. Let $\mathbb{Z}_+ = \mathbb{Z} \cap [0, \infty)$.

Examples 1.1 (Cellular automata) 1. On the set of integers: Let $\mathbb{C} = \mathbb{Z}$, $N(x) = \{x - 1, x, x + 1\}$. The result of transition at site x is

$$f(\xi)(x) = g(\xi(x - 1), \xi(x), \xi(x + 1))$$

where $g(x, y, z)$ is a *local transition function*. In this example, the function f depends only on the sequence of values of $\xi \restriction N(x)$: it is *homogenous*. The present paper will not rely on homogeneity.

2. On the set of natural numbers, with *free boundary condition*: Let $\mathbb{C} = \mathbb{Z}_+$, $N(x) = \{x - 1, x, x + 1\}$ for $x > 0$ and $\{0, 1\}$ for $x = 0$. Given are local transition functions $g(x, y, z)$, $g_0(x, y)$. Now the result of transition at site x is $g(\xi(x - 1), \xi(x), \xi(x + 1))$ for $x > 0$ and $g_0(\xi(0), \xi(1))$ for $x = 0$.

⌟

Let us fix an automaton $\mathbf{A}$ as in (1). An arbitrary function $\eta : \mathbb{C} \times \mathbb{Z}_+ \to \mathbb{S}$ is called a *space-time configuration*. This can also be viewed as a sequence $\eta : \mathbb{Z}_+ \to \mathbb{S}^{\mathbb{C}}$ of space-configurations. We will say that η is a *synchronous trajectory* if for all x, t we have $\eta(\cdot, t + 1) = f(\eta(\cdot, t))$, that is

$$\eta(x, t + 1) = f(\eta(\cdot, t))(x). \tag{2}$$

Each site is "updated" every time by the function f (though the update may not change the state).

We are interested in situations when at any one time, only some of the sites are updated. We will say η is an *asynchronous trajectory* if (2) holds for all x, t with $\eta(x, t+1) \neq \eta(x, t)$: each site in η at each time is either updated or left unchanged. From now on, a "trajectory" without qualification will mean an asynchronous trajectory. Let the *update set*

$$U(t, \eta)$$

be the set of sites x with $\eta(x, t+1) \neq \eta(x, t)$. The initial configuration and the update sets $U(t, \eta)$ determine the whole space-time configuration η. For any set A, let

$$\chi(x, A) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{otherwise.} \end{cases}$$

For a space-time configuration $\eta(x, t)$ we define the *effective age* function $\tau(x, t) = \tau(x, t, \eta)$ as

$$\begin{aligned} \tau(x, 0) &= 0, \\ \tau(x, t+1) &= \tau(x, t) + \chi(x, U(t, \eta)). \end{aligned}$$

This is the number of effective updatings that x underwent until time t. Given an initial configuration ξ, we say that the transition f (and thus the automaton $\mathbf{A}$) has *invariant histories* on ξ if there is a function $\zeta(x, u) = \zeta(x, u, \xi)$ such that for all asynchronous trajectories $\eta(x, t)$ with $\eta(\cdot, 0) = \xi$ we have

$$\eta(x, t) = \zeta(x, \tau(x, t, \eta), \xi). \tag{3}$$

This means that after eliminating repetitions, the sequence $\zeta(x, 0), \zeta(x, 1), \ldots$ of values that a site x will go through in η does not depend on the update sets, only on the initial configuration (except that the sequence may be finite if there is not an infinite number of successful updates). The update sets influence only the delays in going through this sequence. We say that an automaton has *invariant histories* if it has such on *all* initial configurations.

Remark 1.2 The sequence $\zeta(x, 0), \zeta(x, 1), \ldots$ is a sequence of local states but $\zeta(\cdot, n)$ may not be a space-configuration (global state) that appears at any time in a typical asynchronous trajectory. ⌟

Theorem 1 *It is undecidable whether a one-dimensional cellular automaton* $\mathbf{A}$ *with some state space* $\mathbb{S} = \{0, \ldots, n-1\}$ *for some natural number n, has invariant histories.*

The theorem justifies looking for some extra sufficient condition for the invariant history property. For us, this condition will be monotonicity: updating some sites should never hold up progress at other sites. Let us elaborate. The set of *free* sites x in a configuration ξ is defined by

$$L(\xi) = \{x : f(\xi)(x) \neq \xi(x)\}$$

(the site is called free since its update is not "held up"). For a space-time configuration η, let

$$L(t, \eta) = L(\eta(\cdot, t)).$$

For a configuration ξ and a set E of sites, let

$$f(\xi, E)(x) = \begin{cases} f(\xi)(x) & \text{if } x \in E \\ \xi(x) & \text{otherwise.} \end{cases}$$
$$f(\xi, E, F) = f(f(\xi, E), F).$$

With this notation, we have $f(\xi) = f(\xi, \mathbb{C}) = f(\xi, L(\xi))$. Now we can express the condition that η is an asynchronous trajectory by saying that for every t there is a set U with

$$\eta(\cdot, t+1) = f(\eta(\cdot, t), U), \tag{4}$$

and the condition that η is synchronous by requiring $U(t, \eta) = L(t, \eta)$ for each t. We call a transition rule f *monotonic* if $L(t, \eta) \setminus U(t, \eta) \subseteq L(t+1, \eta)$: updating a site cannot take away the freedom of other sites. We call a transition rule f (and thus the automaton **A**) *commutative* if for all configurations ξ and all disjoint sets of sites $A, B \subseteq L(\xi)$ we have

$$f(\xi, A, B) = f(\xi, A \cup B). \tag{5}$$

We call f *locally commutative* when this property is required just for the special case where A, B are one-element sets. The following fact shows that commutativity is locally checkable. It is easy to see, but we give the proof for completeness.

Proposition 1.3 *If the transition function f has finite neighborhoods then its local commutativity implies commutativity.*

Proof. Let us first show

$$f(\xi, \{x_1\}, \dots, \{x_n\}) = f(\xi, \{x_1, \dots, x_n\}). \tag{6}$$

Local commutativity implies for each k,

$$\xi' = f(\xi, \{x_1\}, \dots, \{x_n\}) = f(\xi, \{x_k\}, \{x_1\}, \dots, \{x_{k-1}\}, \{x_{k+1}\}, \dots, \{x_n\}).$$

Therefore $\xi'(x_k) = f(\xi, \{x_1, \dots, x_n\})(x_k)$. Now, let us show

$$f(\xi, \{x_1, \dots, x_n\}, \{y\}) = f(\xi, \{x_1, \dots, x_n, y\}). \tag{7}$$

Using (6), we have $f(\xi, \{x_1\}, \dots, \{x_n\}) = f(\xi, \{x_1, \dots, x_n\})$, hence $f(\xi, \{x_1, \dots, x_n\}, \{y\}) = f(\xi, \{x_1\}, \dots, \{x_n\}, \{y\})$. Using (6) again concludes the proof.

Let us return to the general case. Obviously, it is sufficient to check (5) for sites $y \in B$. Clearly, $f(\xi, A, B)(y) = f(\xi, N(y) \cap A, \{y\})$. The latter is $f(\xi, (N(y) \cap A) \cup \{y\})$ according to (7). □

Remarks 1.4 1. For the cellular automaton example above, local commutativity is equivalent to saying that if $g(r_0, r_1, r_2) \neq r_1$ and $g(r_1, r_2, r_3) \neq r_2$ then

$$g(g(r_0, r_1, r_2), r_2, r_3) = g(r_1, r_2, r_3)$$
$$g(r_0, r_1, g(r_1, r_2, r_3)) = g(r_0, r_1, r_2).$$

2. If f does not have finite neighborhoods then local commutativity does not always imply commutativity. For an example, let $\mathbb{C} = \{0, 1\}$, $\mathbb{C} = \mathbb{Z}$, $N(x) = \mathbb{C}$, and let

$$f(\xi)(x) = \begin{cases} 1 & \text{if } \xi(y) = 0 \text{ for all but finitely many } y, \\ 0 & \text{otherwise.} \end{cases}$$

Now f is obviously locally commutative. On the other hand, let $\xi_0(x) = 0$ for all x, and let Then $f(\xi_0, \mathbb{Z})(-1) = 1$ and $f(\xi_0, \mathbb{Z}_+, \mathbb{Z} \setminus \mathbb{Z}_+)(-1) = 0$.

⌟

Theorem 2 *A transition function is commutative if and only if it is monotonic and has invariant histories.*

It seems that Theorem 2 has been discovered in various frameworks several times, since the invariant histories property is similar to the property of "confluence" in term rewriting (see for example [2]), and to properties of the "sandpile" models in statistical physics. However, the present context probably justifies a self-contained proof.

Theorem 3 *Let $\mathbf{A}_1 = (\mathbb{C}, \mathbb{S}_1, N, f_1)$ be an arbitrary (not necessarily commutative) automaton with finite neighborhoods $N(x)$. Then there is a locally commutative automaton $\mathbf{A}_2 = (\mathbb{C}, \mathbb{S}_1 \times R, N, f_2)$, where for $s \in \mathbb{S}_1 \times R$ we write $s = (s.\mathsf{F}, s.\mathsf{G})$, with the following property.*

Let ξ_1 be an arbitrary configuration of f_1 and let ξ_2 be a configuration of f_2 such that for all x we have $\xi_2(x) = (\xi_1(x), 0 \cdots 0)$. Then for the synchronous trajectory η_2

of f_2, with initial configuration ξ_2, the space-time configuration $(x, t) \mapsto \eta_2(x, t)$. F is a synchronous trajectory of f_1. Moreover, in this trajectory, the state of each cell changes in each step.

In other words, as long as we update synchronously the rule f_2 behaves in its field F just like the arbitrary rule f_1. But f_2 has invariant histories: it is more robust.

2 Commutativity implies invariant histories

Lemma 2.1 *Suppose that f has invariant histories and is monotonic: then it is commutative.*

Proof. Let $U_1(0) = U_2(1) = \{x\}$, $U_1(1) = U_2(0) = \{y\}$, and $U_1(t + 2) = U_2(t + 2)$. This defines η_1 and η_2 from initial configuration ξ by U_1, U_2 as usual. By monotonicity, $\eta_1(y, 1) \neq \eta_1(y, 2)$ and $\eta_2(x, 1) \neq \eta_2(x, 2)$, so τ's values satisfy

$$\tau(x, 2, \eta_1) = \sum_{t=0}^{1} \chi(w, U_1(t, \eta_1))$$

which is 1 if $w \in \{x, y\}$ and 0 otherwise. The same value is obtained for $\tau(x, 2, \eta_2)$. By invariant histories, there is a ζ such that

$$\eta_1(w, 2) = \zeta(w, \tau(w, 2, \eta_1)) = \zeta(w, \tau(w, 2, \eta_2)) = \eta_2(w, 2)$$

and

$$\begin{aligned} f(\xi, x, y) &= f(\xi, U_1(0), U_1(1)) \\ &= \eta_1(w, 2) = \eta_2(w, 2) = f(\xi, U_2(0), U_2(1)) = f(\xi, \{y\}, \{x\}). \end{aligned}$$

Thus, f is commutative. □

What remains to prove after Lemma 2.1 is that commutativity implies monotonicity and invariant histories.

Lemma 2.2 *If f is commutative then it is monotonic.*

Proof. By commutativity $f(\xi, U(t, \eta), L(t, \eta) \setminus U(t, \eta)) = f(\xi, L(r, \eta))$. Therefore $L(t, \eta) \setminus U(t, \eta) \subseteq L(t, \eta)$ implies that f is monotonic. □

We say for two asynchronous trajectories η_0, η_1 with the same initial configuration that η_1 *dominates* η_0 until time u if the following conditions hold:

a) $\tau(\cdot, t, \eta_0) \leq \tau(\cdot, t, \eta_1)$ for all $t \leq u$

b) for all $t_0, t_1 \leq u$, if $\tau(x, t_0, \eta_0) = \tau(x, t_1, \eta_1)$ then $\eta_0(x, t_0) = \eta_1(x, t_1)$.

When η_1 dominates η_0 up to time u for all u then we simply say that η_1 dominates η_0. Domination is, of course, a transitive relation. If the rule has invariant histories then condition (a) implies (b), but otherwise this may not be the case.

Proof of Theorem 2. Let f be a commutative transition rule. It remains to prove that it has invariant histories.

1. Let η be an asynchronous trajectory and $A_0 \subseteq L(0, \eta) \setminus U(0, \eta)$. Then there is an asynchronous trajectory η' dominating η with initial configuration $\eta(\cdot, 0)$ and $U(0, \eta') = U(0, \eta) \cup A_0$.

   Proof. Let $\xi_0 = \eta(\cdot, 0)$. We build, for each u, a trajectory η' with the given properties that dominates η up to time u. When $u \to \infty$ then η' will converge to a trajectory with the same properties that dominates η. For $u = 0$ we can choose $\eta'(\cdot, 0) = \eta(\cdot, 0)$. We assume that η' can be constructed for all $v < u$ and prove it for u. Let $\xi_1 = \eta(\cdot, 1)$, and $A_1 = A_0 \setminus U(1, \eta)$. Let the trajectory η_1 be defined by $\eta_1(x, t) = \eta(x, t + 1)$. The inductive assumption gives a trajectory η'_1 with initial configuration ξ_1 dominating η_1, with

$$U(0, \eta'_1) = A_1 \cup U(0, \eta_1). \tag{8}$$

   Using this trajectory define, for $t > 0$:

$$\eta'(\cdot, t) = \begin{cases} f(\xi_0, A_0 \cup U(0, \eta)) & \text{if } t = 1, \\ \eta'_1(\cdot, t - 1) & \text{otherwise.} \end{cases}$$

   1.1. η' is an asynchronous trajectory.

   Proof. Let us show that η' satisfies (4). This holds by definition for $t = 0$ and $t > 1$. Let us show that it also holds for $t = 1$ with $U = U(1, \eta) \setminus A_0$. We have

$$\begin{aligned} \eta'(\cdot, 2) &= \eta'_1(\cdot, 1) && \text{by def.,} \\ &= f(\xi_1, A_1 \cup U(0, \eta_1)) && \text{by (8),} \\ &= f(\xi_1, A_1 \cup U(1, \eta)) && \text{by def. of } \eta_1, \\ &= f(\xi_1, (A_0 \setminus U(1, \eta)) \cup U(1, \eta)) && \text{by def. of } A_1, \\ &= f(\xi_1, A_0 \cup (U(1, \eta) \setminus A_0)) && \\ &= f(\xi_0, U(0, \eta), A_0, U(1, \eta) \setminus A_0) && \text{by def. of } \xi_1 \\ & && \text{and commutativity,} \\ &= f(\eta'(\cdot, 1), U(1, \eta) \setminus A_0). && \end{aligned} \tag{9}$$

   For domination, we must check two properties.

1.2. We have $\tau(x, t, \eta) \leq \tau(x, t, \eta')$.

Proof. By the definition of τ, for $t > 0$,

$$\tau(x, t, \eta) = \begin{cases} \chi(x, U(0, \eta)) & \text{if } t = 1, \\ \tau(x, 1, \eta) + \tau(x, t-1, \eta_1) & \text{if } t > 1. \end{cases}$$

By the definition of η_1', η, for $t > 0$, using (9), we have

$$\tau(x, 1, \eta_1') = \chi(x, A_1 \cup U(1, \eta)) = \chi(x, A_0 \cup U(1, \eta)). \tag{10}$$

Further,

$$\tau(x, t, \eta') = \begin{cases} \chi(x, A_0 \cup U(0, \eta)) & \text{if } t = 1, \\ \tau(x, 1, \eta') + \chi(x, U(1, \eta) \setminus A_0) & \text{if } t = 2, \\ \tau(x, 2, \eta') + \tau(x, t-1, \eta_1') - \tau(x, 1, \eta_1') & \text{if } t > 2. \end{cases} \tag{11}$$

By the above definition,

$$\begin{aligned} \tau(x, 1, \eta') &= \tau(x, 1, \eta) + \chi(x, A_0), \\ \tau(x, 2, \eta') &= \tau(x, 1, \eta) + \chi(x, A_0) + \chi(x, U(1, \eta) \setminus A_0) \\ &= \tau(x, 1, \eta) + \chi(x, A_0 \cup U(1, \eta)) \geq \tau(x, 1, \eta) + \chi(x, U(1, \eta)) \\ &= \tau(x, 2, \eta). \end{aligned}$$

Also, from here and (10),

$$\tau(x, 2, \eta') = \tau(x, 1, \eta) + \chi(x, A_0 \cup U(1, \eta)) = \tau(x, 1, \eta) + \tau(x, 1, \eta_1'). \tag{12}$$

By domination, $\tau(x, t-1, \eta_1') \geq \tau(x, t-1, \eta_1)$ and hence for all $t \geq 2$, we have, combining (11) with (12),

$$\begin{aligned} \tau(x, t, \eta') &= \tau(x, 1, \eta) + \tau(x, t-1, \eta_1') \\ &\geq \tau(x, 1, \eta) + \tau(x, t-1, \eta_1) = \tau(x, t, \eta). \end{aligned} \tag{13}$$

1.3. If $\tau(x, s, \eta) = \tau(x, s', \eta')$ then $\eta(x, s) = \eta'(x, s')$.

Proof. If $\tau(x, s, \eta) = 0$ then clearly $\eta'(x, s) = \eta'(x, s')$ since this means that in both processes, no progress has been made in x from the initial configuration. Assume therefore $\tau(x, s, \eta) > 0$ and hence $s, s' > 0$.

Assume $s' = 1$. Then $\tau(x, s, \eta) = \tau(x, 1, \eta') = 1$ and hence $x \in A_0 \cup U(0, \eta)$. If $x \in U(0, \eta)$ then $s = 1$ and hence the same transition that gives $\eta'(x, 1)$

also gives $\eta(x, 1)$. Otherwise $s > 1$ hence $\tau(x, s-1, \eta_1) = 1$. Also, $x \in A_0 \subseteq U(0, \eta_1')$, hence $\tau(x, 1, \eta_1') = 1$. The inductive assumption implies $\eta_1'(x, 1) = \eta_1(x, s-1) = \eta(x, s)$. On the other hand, (9) and $x \notin U(0, \eta)$ implies $\eta_1'(x, 1) = \eta'(x, 1)$ which concludes this case.

Assume now $s' > 1$. Since $\eta(x, t)$ changes if and only if $\tau(x, t)$ does we can assume that $x \in U(s, \eta)$ since otherwise we can decrease s without changing $\eta(x, s)$. The same is true for s'. Under these assumptions we have $s \geq s'$. By (13),

$$\tau(x, s', \eta') = \tau(x, 1, \eta) + \tau(x, s'-1, \eta_1').$$

We assumed this to be equal to $\tau(x, s, \eta) = \tau(x, 1, \eta) + \tau(x, s-1, \eta_1)$. Hence $\tau(x, s'-1, \eta_1') = \tau(x, s-1, \eta_1)$. Also $\eta(x, s) = \eta_1(x, s-1)$, $\eta'(x, s') = \eta_1'(x, s'-1)$, and hence the inductive assumption implies the statement.

2. Let η be a trajectory. Then the synchronous trajectory with initial configuration $\eta(\cdot, 0)$ dominates η.

Proof. Let $A_0 = L(0, \eta) \setminus U(0, \eta)$. By 1 above, there is a trajectory η' with initial configuration $\eta(\cdot, 0)$ dominating η such that $U(0, \eta') = U(0, \eta) \cup A_0 = L(0, \eta)$. This just means that η' is a synchronous trajectory up to time 1. Continuing the application of 1, we can dominate η by a synchronous trajectory η'' up to time 2, and so on.

Now we can conclude the proof of the theorem as follows. Let η be a trajectory with initial configuration ξ and let η' be the synchronous trajectory with the same initial configuration. Let us define

$$\begin{aligned}\sigma(x, s, \xi) &= \min\{t : \tau(x, t, \eta') = s\},\\ \zeta(x, s, \xi) &= \eta'(x, \sigma(x, s)).\end{aligned}$$

To prove (3), note that due to domination, $\tau(x, t, \eta) \leq \tau(x, t, \eta')$ and hence for every x, y, t there is a $t' \leq t$ with $\tau(x, t, \eta) = \tau(x, t'\eta')$. Let t' be the first such: $t' = \sigma(s, \tau(x, t, \eta))$. By domination, $\eta(x, t) = \eta'(x, t') = \zeta(x, t)$. □

3 A rich example of commutative transitions

In this section, we will prove Theorem 3.

We will use the following notation:

$$b \text{ amod } m$$

is the integer x with $x \equiv b \pmod{m}$ and $-m/2 < x \leq m/2$.

Proof. Let $\mathbb{S}_2 = \mathbb{S}_1^2 \times \{0, 1, 2\}$. The three components of each state s of $\mathbb{S}_2$ will be written as

$$s.\,\mathsf{Cur}\,, s.\,\mathsf{Prev} \in \mathbb{S}_1,\;\; s.\,\mathsf{Age} \in \{0, 1, 2\}.$$

The statement of the theorem will obtain by $s.\,\mathsf{F} = s.\,\mathsf{Cur}$, $s.\,\mathsf{G} = (s.\,\mathsf{Prev}\,, s.\,\mathsf{Age})$. The field $\mathsf{Age} \in \{0, 1, 2\}$ will be used to keep track of the time of the simulated cells mod 3, while Prev holds the value of Cur for the previous value of Age.

Define $s' = f_2(\xi)(x)$. If there is a $y \in N(x)$ with $(\xi(y).\,\mathsf{Age} - \xi(x).\,\mathsf{Age})$ amod $3 < 0$ (that is some neighbor lags behind) then $s' = \xi(x)$, there is no effect. Otherwise, let $\sigma(y)$ be $\xi(y).\,\mathsf{Cur}$ if $\xi(y).\,\mathsf{Age} = \xi(x).\,\mathsf{Age}$, and $\xi(y).\,\mathsf{Prev}$ otherwise.

$$\begin{aligned} s'.\,\mathsf{Cur} &= f_1(\sigma)(x), \\ s'.\,\mathsf{Prev} &= \xi(x).\,\mathsf{Cur}\,, \\ s'.\,\mathsf{Age} &= \xi(x).\,\mathsf{Age} + 1 \bmod 3. \end{aligned}$$

Thus, we use the Cur and Prev fields of the neighbors according to their meaning and update the three fields according to their meaning. It is easy to check that this transition rule simulates f_1 in the Cur field if we start it by putting 0 into all other fields.

Let us check that f_2 is locally commutative. If two neighbors x, y are both are allowed to update then neither of them is behind the other modulo 3, hence they both have the same Age field. Suppose that x updates before y. In this case, x will use the the Cur field of y for updating and put its own Cur field into Prev. Next, since now x is “ahead” according to Age, cell y will use the Prev field of x for updating: this was the Cur field of before. Therefore the effect of consecutive updating is the same as that of simultaneous updating. □

The commutative medium of the above proof is also called the *marching soldiers* scheme since its handling of the Age field reminds one of a chain of soldiers marching ahead in which two neighbors do not want to be separated by more than one step. It is shown in [1] that if the update times obey a Poisson process then the average computation time of this simulation within a constant factor of the computation time of the synchronous computation.

Remark 3.1 In typical cases of asynchronous computation, there are more efficient ways to build a commutative rule than to store the whole previous state in the Prev field. Indeed, the transition function typically does not use the complete state of cells in $N(x)$. Rather, the cells only “communicate” in the sense that there is a message field and the next state of x depends only on this field of the neighbor cells. In such cases, it is sufficient in the above construction to store the previous value of this message field. We can sometimes decrease the message field by taking several steps of f_2 to simulate a single step of f_1. ⌟

In case of one-dimensional systems, as in Example 1.1, the "marching soldiers" scheme has the following strengthening, saying that *every* asynchronous trajectory η codes a synchronous computation, no matter what its initial configuration:

Theorem 4 *For an arbitrary one-dimensional cellular automaton* $\mathbf{A}_1 = (\mathbb{C}, \mathbb{S}_1, N, f_1)$ *given, as in Example 1.1, via a local transition function* g*, define automaton* $\mathbf{A}_2 = (\mathbb{C}, \mathbb{S}_1 \times R, N, f_2)$ *as in the proof of Theorem 3. For an arbitrary asynchronous trajectory* η *of* $\mathbf{A}_2$*, define the "delay function"* $\delta(x)$ *and the "straightened" space-time configuration* $\bar\eta(x, u)$*, as follows. Let* $\delta(0) = 0$*, and*

$$\begin{aligned}\delta(x+1) &= \delta(x) + \eta(x+1, 0).\mathsf{Age} - \eta(x, 0).\mathsf{Age} \quad \textit{(not reducing modulo 3)},\\ \bar\tau(x, t) &= \tau(x, t) + \delta(x),\\ \bar\eta(x, u) &= \zeta(x, u - \delta(x)).\mathsf{Cur}\end{aligned}$$

for all u *of the form* $\bar\tau(x, t)$*. Also, let* $\bar\eta(x, \delta(x) - 1) = \eta(x, 0).\mathsf{Prev}$*. Then* $\tau(x, t) > 0$ *implies with* $u = \bar\tau(x, t) - 1$ *that*

$$\bar\eta(x, u+1) = g(\bar\eta(x-1, u), \bar\eta(x, u), \bar\eta(x+1, u)),$$

and all terms in this equation are defined.

The proof is straightforward verification. The synchronous trajectory of $\mathbf{A}_1$ derived from the asynchronous trajectory η of $\mathbf{A}_2$, is $\bar\eta(x, u) = \zeta(x, u - \delta(x)).\mathsf{Cur}$. The delay function $\delta(x)$ shows how much "ahead" or "behind" is $\eta(\cdot, 0)$ in simulating the synchronous trajectory.

Remark 3.2 This theorem fails in in networks containing cycles: there, only certain initial configurations $\eta(\cdot, 0)$ allow the construction of the delay function $\delta(x)$. In the ones that do not allow it, there is some inconsistency in the timing function $\eta(x, 0).\mathsf{Age}$ (a loop along which the sum of local increments of Age is not 0). In a connected network, this loop will imply that each cell can have only finitely many state changes, even in an infinite trajectory. ⌟

4 Undecidability

This section proves Theorem 1.

Lemma 4.1 *Consider one-dimensional commutative cellular automata with sites on the set of natural numbers, with free boundary condition, as in Example 1.1.2 by a set of states* $\mathbb{S} = \{0, \ldots, n-1\}$*, transition functions* $g : \mathbb{S}^3 \to \mathbb{S}$ *and* $g_0 : \mathbb{S}^2 \to \mathbb{S}$*, with* $g(0, 0, 0) = 0$*,* $g_0(1, s) = 1$ *(for all* s*). The following problem is undecidable, as a function of* n, g, g_0*: Is there any synchronous trajectory of this cellular automaton, with* $\eta(x, 0) = 0$ *for all* x *and* $\eta(0, t) = 1$ *for some* $t > 0$*?*

Proof. There is a standard construction to simulate Turing machines with such cellular automata, so the question reduces to the question whether an arbitrary Turing machine will halt when started on an empty tape. □

Lemma 4.2 *Consider one-dimensional commutative cellular automata over the set of natural numbers, with free boundary condition, set of states* $\mathbb{S} = \{0, \ldots, n-1\}$*, transition functions* $g : \mathbb{S}^3 \to \mathbb{S}$ *and* $g_0 : \mathbb{S}^2 \to \mathbb{S}$ *as in Example 1.1.2.*

The following problem is undecidable, as a function of n, g, g_0*: Is there any asynchronous trajectory of this cellular automaton, with* $\eta(0, 0) = 0$ *and* $\eta(0, t) = 1$ *for some* $t > 0$*?*

The main difference between this lemma and the previous one is that we do not require the initial configuration $\eta(x, 0)$ to be 0 for all x, only for $x = 0$. Otherwise, since the automaton is commutative it does not matter whether the trajectory asked for is synchronous or asynchronous.

Proof. From now on, without danger of confusion, let us write $g(r, s) = g_0(r, s)$ and forget about g_0. Let us be given a cellular automaton g like in Lemma 4.1, with state set $\mathbb{S} = \{0, \ldots, n-1\}$. We construct a new cellular automaton over the set of states $\mathbb{S}' = \mathbb{S} \cup \{n\}$, with the following transition function g'. Over states $s < n$, the functions g' behave as g. Further, we have the following rules for g' when at least one of the arguments is n.

$$
\begin{aligned}
(n, s) &\mapsto g(0, 0), \\
(s, n) &\mapsto g(s, 0) && \text{for } s < n, \\
(n, r, s) &\mapsto n, \\
(r, n, s) &\mapsto g(r, 0, 0) && \text{for } r < n, \\
(r, s, n) &\mapsto g(r, s, 0) && \text{for } r, s < n,
\end{aligned}
$$

and $(r, s, n) \mapsto s$, $(r, s) \mapsto r$ in all remaining cases. By these rules, the symbol n "sweeps" right and in its wake the rule g will operate as if it had started from the a configuration of all 0's. Thus, let η be the synchronous trajectory of g with $\eta(x, 0) = 0$ for all x. Then clearly if η' is any synchronous trajectory of g' with $\eta'(0, 0) = n$ then for all $t > 0$, for all $x \le t$ we have $\eta'(x, t) = \eta(x, t)$.

Let us now apply the construction of the proof of Theorem 3 to g' to obtain commutative rule g'' over the set of states $\mathbb{S}'' = (\mathbb{S}')^2 \times \{0, 1, 2\}$. We will prove that g'' has an asynchronous trajectory η'' with $\eta''(0, 0) = (n, 0, 0)$ and $\eta''(0, u) = (1, 1, 0)$ for some u, if and only if g has a synchronous trajectory η with $\eta(0, x) = 0$ for all x and $\eta(0, u) = 1$ for some u. Since we know that the question whether this happens is undecidable from g, we will have proved that the question whether some cellular automaton has an asynchronous trajectory η with $\eta(0, 0) = s_0$ and $\eta(0, u) = s_1$ for some $s_0 \neq s_1$ is undecidable; this will complete the proof.

The "if" part: Suppose first that g has a synchrounous trajectory η with $\eta(0, x) = 0$ for all x, and and $\eta(0, u) = 1$ for some u. As mentioned above, then the synchronous trajectory η' of g' has $\eta'(x, t) = \eta(x, t)$ for all $x \leq t$. Consider the synchronous trajectory η'' of g'' started from $\eta''(x, 0) = (n, 0, 0)$ for all x. Then for all $t > 0$ and all $x \leq t$ we have

$$\eta''(x, t) = (\eta'(x, t), \eta'(x, t-1), t \bmod 3) = (\eta(x, t), \eta(x, t-1), t \bmod 3).$$

Let v be the first number $> u + 1$ divisible by 3. We have

$$\eta''(0, v) = (\eta(0, v), \eta(0, v-1), 0) = (1, 1, 0).$$

The "only if" part: Assume that η'' is an asynchronous trajectory of g'' with $\eta''(0, 0) = (n, 0, 0)$ and $\eta''(0, w) = (1, 1, 0)$ for some w. Then $\tau''(0, w) > 0$ and defining $u = \bar{\tau}''(0, w) - 1$, Theorem 4 implies

$$\bar{\eta}''(0, u + 1) = g'(\bar{\eta}''(0, u), \bar{\eta}''(1, u)).$$

Applying the theorem repeatedly, we obtain

$$\bar{\eta}''(0, v + 1) = g'(\bar{\eta}''(x - 1, v), \bar{\eta}''(x, v), \bar{\eta}''(x + 1, v))$$

or, if $x = 0$, the same relation with the first argument of g' omitted, for $v = 0, \dots, u$ and $x \leq \min\{v, (u-v)\}$. Now, if $\eta''(0, w) = (1, 1, 0)$ then $\bar{\eta}''(0, u+1) = 1$ while $\bar{\eta}''(0, 0) = n$. We have just found that $\bar{\eta}''(x, v)$ develops according to g' for $v = 0, \dots, u$ and $x \leq \min\{v, (u-v)\}$. As discussed above, therefore $\bar{\eta}''(0, u+1) = 1$ if and only if g computes 1 at $(0, u + 1)$ from an all-0 initial configuration. $\square$

Proof of Theorem 1. Let the local state space be the set of integers $\mathbb{S} = \{0, \dots, n + 2\}$. Let $g : \mathbb{S}_0^3 \to \mathbb{S}_0$ and $g_0 : \mathbb{S}_0^2 \to \mathbb{S}_0$ be the rules for a commutative cellular automaton transition rule with state set $\mathbb{S}_0 = \{0, \dots, n-1\}$. We define the transition function f. We will write $f(x, y, z) = y'$ as $(x, y, z) \mapsto y'$. We require

$$(s, n, 0) \mapsto n + 1, \tag{14}$$

$$(s, n, 1) \mapsto n + 2, \tag{15}$$

$$(r, s, t) \mapsto g_0(s, t) \qquad \text{for all } r \geq n,\ r, s < n,\ r \neq 1, \tag{16}$$

$$(r, s, t) \mapsto g(r, s, t) \qquad \text{for all } r, s, t < n, \tag{17}$$

$$(r, s, t) \mapsto g(r, s, 0) \qquad \text{for all } r, s < n,\ t \geq n, \tag{18}$$

and $(r, s, t) \mapsto s$ in all remaining cases. Let us show that f has invariant histories if and only if g has no asynchronous trajectory η_0 over $\mathbb{C} = \mathbb{Z}_+$ with $\eta_0(0, 0) = 0$

and $\eta_0(0, t) = 1$ for some t. Assume first that g has such a trajectory. Let us define the initial configuration ξ of f as $\xi(x) = n$ if $x = -1$ and 0 otherwise. We may apply rule (14) first to get $\eta(-1, 1) = n + 1$. Or, we may apply rules (16),(17),(18) first to cells $x > 0$ on the right repeatedly. Sooner or later we have $\eta(0, t) = 1$, which allows $\eta(-1, t + 1) = n + 2$ by rule (15) in the next step. Thus, depending on the order of rule application, we obtained in cell -1 the sequence $n, n + 1$ or $n, n + 2$.

Suppose now that g has no such trajectory and let ξ be an arbitrary configuration of f. Each occurrence of a state $\geq n$ remains such an occurrence. On segments between them, the commutative rule g works. The only other transitions possible are $(r, n, 0) \mapsto n + 1$ and $(r, n, 1) \mapsto n + 2$. Assume $\eta(x, 0) = n$ and consider the sequence of different values in $\eta(x + 1, t)$. Let us show that 0 and 1 cannot both occur in this sequence and hence only one of the transitions is possible. Indeed, if 0 occurs before 1 then our assumption about g excludes the occurrence of 1 in the sequence any later. If 1 occurs in the sequence before 0 then our rules (in particular (16)) do not allow any change of the state of $x + 1$ after that. $\square$

Acknowledgment I thank Robert Solovay for pointing out several errors in the first version, Wayne Snyder for calling my attention to the confluence literature and Deepak Dar for pointing to the sandpile literature.

Bibliography

[1] Piotr Berman and Janos Simon. Investigations of fault-tolerant networks of computers. In *Proc. of the 20-th Annual ACM Symp. on the Theory of Computing*, pages 66–77, 1988. 3

[2] Gerard Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *J. of the Assoc. for Computing Machinery*, 27(4):797–821, October 1980. 1