When – and how – can a cellular automaton be rewritten as a lattice gas?

Tommaso Toffoli [a,*], Silvio Capobianco [b], Patrizia Mentrasti [c]

[a] *Electrical and Computer Engineering, Boston University, United States*
[b] *School. Comp. Sci., Reykjavik University, Iceland*
[c] *Dip. di Matematica, Università di Roma "La Sapienza", Italy*

A R T I C L E   I N F O

A B S T R A C T

Both *cellular automata* (CA) and *lattice-gas automata* (LG) provide finite algorithmic presentations for certain classes of infinite dynamical systems studied by symbolic dynamics; it is customary to use the terms 'cellular automaton' and 'lattice gas' for a dynamic system itself as well as for its presentation. The two kinds of presentation share many traits but also display profound differences on issues ranging from decidability to modeling convenience and physical implementability.

Following a conjecture by Toffoli and Margolus, it had been proved by Kari that any *invertible* CA, at least up to two dimensions, can be rewritten as an isomorphic LG. But until now it was not known whether this is possible in general for *noninvertible* CA—which comprise "almost all" CA and represent the bulk of examples in theory and applications. Even circumstantial evidence – whether in favor or against – was lacking.

Here, for noninvertible CA, (a) we prove that an LG presentation is out of the question for the vanishingly small class of *surjective* ones. We then turn our attention to all the rest – noninvertible *and* nonsurjective – which comprise all the typical ones, including Conway's 'Game of Life'. For these (b) we prove by explicit construction that all the one-dimensional ones are representable as LG, and (c) we present and motivate the conjecture that this result extends to *any* number of dimensions.

The tradeoff between dissipation rate and structural complexity implied by the above results have compelling implications for the thermodynamics of computation at a microscopic scale.

© 2008 Elsevier B.V. All rights reserved.

I do not know of any single instance where something useful for the work on lattice gases has been borrowed from the cellular automata field. . . .. Lattice gases differ in essence from cellular automata. A confusion of the two fields distorts our thinking, hides the special properties of lattice gases, and makes it harder to develop a good intuition. (Michel Hénon [16], with specific reference to Wolfram [41])

1. Introduction

Cellular automata (CA) provide a quick modeling route to phenomenological aspects of nature—especially the emergence of complex behavior in *dissipative* systems. But lattice-gas automata (LG) are unmatched as a source of fine-grained models of fundamental aspects of physics, especially for expressing the dynamics of *conservative*[1] systems.

---

* Corresponding author.
E-mail addresses: tt@bu.edu (T. Toffoli), silvio@ru.is (S. Capobianco), mentrasti@mat.uniroma1.it (P. Mentrasti).

[1] A dynamics is called 'conservative' if it is the manifest expression of an *invertible* microscopic mechanism. It is called 'dissipative' if the underlying mechanism is not invertible to begin with, or if its invertibility is de facto irrelevant because one is not capable, or willing, to maintain a strict accounting of microscopic states—perhaps owing to lack of precise knowledge of the initial state and the evolution laws, unpredictable influences on the part of the environment, or the sheer size of the task.

In the quote at the beginning of this paper, one may well sympathize with Hénon's annoyance: it turns out that dynamical behavior that is synthesized with the utmost naturalness when using lattice gases as a "programming language" become perversely hard to express in the cellular automata language. Yet, Hénon's are visceral feelings, not argued conclusions. With as much irritation one could retort, "How can lattice gases differ 'in essence' from cellular automata if they are merely a subset of them? What are these CA legacies that may 'distort our thinking' and 'hide the special properties of lattice gases'? And aren't there dynamical systems that are much more naturally and easily modeled as cellular automata"?

Today, with the benefit of twenty years' hindsight – and especially after the results of the present paper – we are in a position to defuse the argument. Hénon's appeal could less belligerently be reworded as follows: "Even though CA and LG describe *essentially the same class* of objects, for sound pedagogical reasons it may be expedient to deal with them in separate chapters—or even in separate books for different audiences and applications. What is ox in the stable may well be beef on the table".

The bottom line is that these two modeling approaches do not reflect mutually exclusive strategies, but just opposite tradeoffs between the structural complexity of a piece of computing machinery and its thermodynamic efficiency. By casting essential aspects of dynamics in a precise formal context, it becomes possible to explicitly show *why*

- *Total* recycling of information waste can in principle be achieved even in *noninvertible* dynamics.
- At the same time, while waste is so easy to produce with simple machinery operating on a local scale, effective recycling of waste may not be possible without *very complex* machinery insuring coordination on a *wide-range scale*. "Logic for capitalists"? (cf. [10].)

2. Preview

Let $\mathcal{C}$ and $\mathcal{L}$ be the sets of dynamical systems representable, respectively, as cellular automata and lattice gases. Our overall question is, How are these two sets related? On one hand, we shall see that any lattice gas can be trivially rewritten as a cellular automaton—hence $\mathcal{L} \subset \mathcal{C}$. As for the converse issue, i.e., *how much of $\mathcal{C}$ is in $\mathcal{L}$*, let's recall that $\mathcal{C}$ is naturally partitioned into three classes which reflect fundamental categorical properties, namely,

	descriptive name	categorical property
$\mathcal{C}^+$	invertible	injective and surjective
$\mathcal{C}^\circ$	almost invertible	noninjective but surjective
$\mathcal{C}^-$	locally lossy	noninjective, nonsurjective

According to a conjecture made by one of us in 1990 and proved by Kari in 1996 for up to two dimensions, the vanishingly small class $\mathcal{C}^+$ of *invertible* cellular automata is definitely "in". For the other two classes, evidence has been lacking either way[2] —which is particularly irritating for such deceptively simple CA as the one consisting merely of a row of 2-input AND gates.

Here we first dismiss, as definitely "out" for *any* number of dimensions, the vanishingly small class $\mathcal{C}^\circ$ of almost invertible CA—where, while the system as a whole loses in one step a nonzero amount of information, the amount of loss *per site* is nonetheless zero.

What is left is the class $\mathcal{C}^-$ of locally lossy CA; these lose in one step a nonzero amount of information per site. This class comprises *almost all* cellular automata. Here we present a procedure for rewriting any one-dimensional $\mathcal{C}^-$ CA into an LG (which will have a more complex unit cell, consisting of two layers and spanning more than one site of the original CA). We also show a procedure for a simple 2-D example. Work is in progress on an analogous construction (with a number of layers that increases with the number of dimensions) hoped to be of general applicability.

On the basis of the above evidence, we propose the conjecture that – with the exception made for the paradoxical class $\mathcal{C}^\circ$ already dismissed above – *the family of dynamical systems presentable as cellular automata and that presentable as lattice gases* coincide.

3. Background

Symbolic dynamics studies a class of dynamical systems which display topological continuity (reflecting locality of interaction) and invariance under space and time translation (the regularity of a "spacetime crystal" [38])—and can thus be viewed as discrete, locally-finite versions of the partial differential equations of field theories. For example, the local map of a simple one-dimensional cellular automaton, which is a recurrence relation of the form

$$q_x^{t+1} = f(q_x^t, q_{x+1}^t)$$

---

[2] Except for *second-order* cellular automata, already proved "in" by us in 2004 (see Section 7.1), and a few other sporadic cases. Of course, for any CA that was obtained from an LG as per Theorem 1, an LG presentation is automatically available.
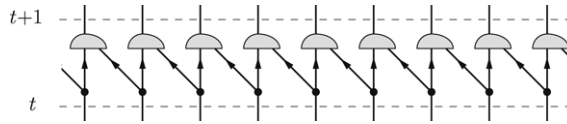
Fig. 1. A simple one-dimensional *cellular automaton*; this circuit can be thought of as a *presentation* of a dynamical system on the full shift [24,25]. For concreteness, assume that the site state alphabet is binary and the local transition function at each site, denoted by a shaded gate, is a logic AND; the gates are fed by *fanout* (or "signal replicator") nodes, denoted by dots. Is there an equivalent lattice gas?
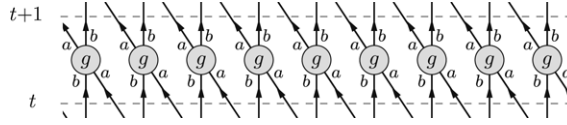

Fig. 2. Format of a simple one-dimensional lattice gas. Here we may imagine *g* to be an arbitrary function of two binary inputs, yielding as a result two distinct binary outputs. No fanout of signals (cf. the fanout nodes of Fig. 1) is permitted. The labels *a* and *b* distinguish the two inputs as well as the two outputs; they actually are labels for *arcs* rather than *I/O ports*.

is analogous to the well-known *forward-time, forward-space* finite-difference scheme [31, p. 13]

$$q_x^{t+1} = g(q_x^t, q_{x+1}^t) = q_x^t + a\frac{k}{h}(q_{x+1}^t - q_x^t)$$

for the differential equation

$$\frac{\partial q}{\partial t} = a\frac{\partial q}{\partial x};$$

see also the approach in [15]. (One must keep in mind that in a cellular automaton the state set for q_x^t is restricted to a finite alphabet A, while for a differential equation it ranges over the whole continuum $\mathbb{R}$. On the other hand, the local map f of a cellular automaton is an unrestricted function of its arguments, while the corresponding map g of the finite-difference scheme can only see its arguments through "small differences"—$q_{x+1} - q_x$, of the order of h, in space, and $q^{t+1} - q^t$, of the order of k, in time.)

Though symbolic dynamics systems are generally infinite, both *cellular automata* and *lattice-gas automata* manage to provide *finite algorithmic presentations* for certain classes of them. These two kinds of presentation share a number of traits but also display *profound differences* – on issues ranging from decidability to physical implementability – to the point that one might suspect that CA and LG specify quite different *kinds* of dynamical systems (see the Hénon quote at the beginning of this paper).

On the other hand, for a number of reasons, some empirical and some aesthetical, we've long entertained the notion – the hope, if you wish, since the evidence was extremely scant either way – that in fact CA and LG just provide different *presentations* for the *same kind of objects*. In other words, letting $\mathcal{C}$ denote the set of dynamical systems representable as cellular automata and $\mathcal{L}$ the set of those representable as lattice gases, one may ask, How are $\mathcal{C}$ and $\mathcal{L}$ related? Are they disjoint, partially overlapping, or even coincident? If the latter were true, how come that LG presentations of even the very simplest CA are so hard to come by?
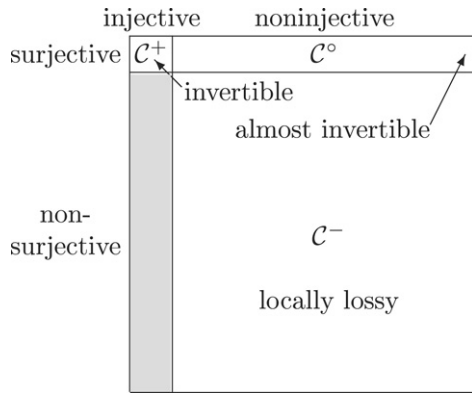
Take, for example, the simplest nontrivial CA, namely, a string of 2-input AND gates fed by fanout nodes (Fig. 1). Until now, no one had managed to exhibit a functionally equivalent LG or, contrariwise, prove that such an LG cannot exist! What's perhaps even more intriguing, this very question has never (as far as we know) been raised in the literature, even though the analogous problem for *invertible* CA had been stated and made the object of a conjecture fifteen years ago [36] and then positively solved in the ensuing decade [18]; and similarly, though more recently, for *second-order* CA [37]. Apparently, everyone was just as clueless about the present simple AND example as we ourselves were until yesterday.

Before proceeding further it will be convenient to informally recall the definition of 'lattice gas' and the structural difference between cellular automata and lattice gases.

The simplest nontrivial[3] LG have the format of Fig. 2. Instead of having, as in a cellular automaton (Fig. 1), a single output from each node and making as many copies of it as necessary to "fan it out" to the nodes that at the next time step will use it as an input, a lattice gas *does not make use of signal fanout*; instead, each node has as many outputs (these may have *different* values, not just copies of the same one) as it has inputs. More precisely, for a CA of state alphabet A and number of neighbors n the local map is of the form $f : A^n \to A$. In an LG, on the other hand, the state alphabet has the structure $A = A_1 \times A_2 \times \cdots \times A_n$ (the factor alphabets $A_1, \ldots, A_n$ need not have the same number of elements), and the local map has the form $g : A_1 \times A_2 \times \cdots \times A_n \to A_1 \times A_2 \times \cdots \times A_n$, with no fanout required or permitted.

As we shall see in Theorem 1, lattice-gas automata can be thought of as a special case of cellular automata. They were used (without being given a special name) by Toffoli in [32,33] and then extensively investigated by Margolus and Toffoli [26, 35,36]; one term used at that time was 'partitioning cellular automata'. In 1985, a rudimentary CA model of fluid dynamics

---

[3] A CA is *trivial* if it has a neighborhood of size 1, and thus, up to a translation, the dynamics decomposes into identical independent dynamics for each site.

Fig. 3. A CA presentation of the system described by the LG of Fig. 2. Both *a* and *b*, bundled together as *q*, are duplicated by the fanout nodes, but then one copy of each (starred) is ignored by g', which in this way effectively acts just as *g*.

discovered independently by the latter research program turned out to be quite similar to one proposed a few years before by the hydrodynamicist Yves Pomeau and colleagues; in turn, this convergence soon stimulated a whole industry of "lattice-gas hydrodynamics" research (see [36] for references). At that time, since the term 'lattice gas' had already been used for sundry discrete models of fluid dynamics, it became customary to call those special cellular automata 'lattice-gas automata' or simply 'lattice gases'; the term 'block cellular automata' is also in common use. The no-fanout constraint is used with the same meaning, though in an independent line of research, in Girard's *linear logic* [14]. There, as here, multiple uses require explicit duplication, with all the costs (in infrastructure or running expenses) that may entail (see also [3]).

The prototypical event in a lattice gas is the collision of abstract tokens or "particles." In many practical cases this may involve a mere reshuffling of them. A generalization of this process is when what is permuted are not the particles but the *states* of the input tuple (a reshuffling of particle is then the special case of a permutation of the *indices* of this tuple); this "transmutation" may be visualized as the creation, out of the collision of particles, of an "excited state" (corresponding to a lattice-gas node), which on decaying releases particles that may differ from the original ones.

Note that in Fig. 1 as well as Fig. 2 the *state* q^t of the system at time *t* is the infinite collection of signals crossing the dashed line *t*. The combinational-logic diagram drawn between the two dashed lines constitutes a *presentation* of the dynamical system, that is, a way to indicate by an explicit, locally-finite algorithm the functional dependency of the state at time $t + 1$ on that at time *t*. Namely, for the CA of Fig. 1, using *i* as a sequential site index, the algorithm specifies that

$$q_i^{t+1} = \text{AND}\,(q_i^t, q_{i+1}^t);$$

different presentations are of course possible for the same system. In the LG of Fig. 2, the state-component at each site consists of a *pair* of signals, labeled *a* and *b*, and the local algorithm specifies that

$$a_i^{t+1} = g_a(a_i^t, b_{i+1}^t) \quad \text{and} \quad b_i^{t+1} = g_b(a_i^t, b_{i+1}^t).$$

Either in a CA or in an LG the presentation may consist of any finite number of combinational-logic *layers*, as in Fig. 5, rather than a single layer as in the previous two figures, each layer obeying the respective discipline for a CA or an LG.

Finally, two dynamical-system presentations are *equivalent* (or *conjugate* [24]) if the systems they describe are isomorphic—the presentations are "merely different views of the same underlying object". Note that, in rewriting a CA as an LG, for the latter to be equivalent to the former in the strict sense used here one is not permitted to introduce supplementary variables, or 'ancillae', to be used as a scratchpad for intermediate computations (cf. [34] and [5]); one must make do with whatever state-variables are already available to each site.

To anticipate Section 12, we mention here that the above distinction between CA and LG makes a vital difference when one turns to the physical implementation of an algorithm. Even in ideal conditions, a fanout node – which creates copies of a signal – needs by its very nature a source of energy to operate; moreover, as Landauer argued in [21], energy is then turned into heat when signals are erased, discarded, or destroyed – as in the AND gates of Fig. 1, which take in two binary lines but only put out one. In this sense, CA provide – for the same functionality – *cheaper mechanisms* than LGs: they are easier to fabricate (as we shall see in a moment), but need a power source to operate.

From the above definitions, one can see that

Theorem 1. *Any LG is immediately rewritable as a CA.*

Proof. We use the LG of Fig. 2 as a specific example, but the construction method is fully general; starting with that LG, we gradually modify it into a CA. Represent the pair of signals $\langle a, b \rangle$ leaving or entering a node as a single signal $q = \langle a, b \rangle$ (if $a \in A$ and $b \in B$, then $q \in A \times B$). With reference to Fig. 3, as soon a signal *q* at site *i* crosses the time line *t* we put it through a fanout element to obtain two copies of it, and feed one copy to the node (embodying a function g') at site *i* and the other to that at $i - 1$. A node receives, through inputs $q_i = \langle a_i, b_i \rangle$ and $q_{i+1} = \langle a_{i+1}, b_{i+1} \rangle$), one *a* input from site $i + 1$ and one *b* input from site *i* – just like a node *g* of Fig. 2 – as well as two extra inputs, namely, one *b* from site $i + 1$ and one *a* from *i*. We define g' as the function which ignores the latter two inputs (starred, in the figure) and otherwise acts identically to *g*. The resulting system is a CA isomorphic to the original LG. ∎

The outcome of this is that, despite the difference in presentation, the functional dependence of the global state at $t + 1$ on that at *t* is the *same* in both Fig. 2 and Fig. 3. Clearly, transcribing from an LG format to a CA one is easy; how hard is the opposite direction?

We shall now draw a rough chart of the area we intend to explore; this area need not extend beyond $\mathcal{C}$ since, by Theorem 1, all of $\mathcal{L}$ (whatever its extent may turn out to be) is contained in $\mathcal{C}$.

$$\tag{1}$$

In diagram (1) the class $\mathcal{C}$ of symbolic dynamics systems presentable as cellular automata is shown partitioned into four subclasses according to whether or not a system is surjective (or "onto") and whether or not it is injective (or "one-to-one"). This classification, which is of a fundamental category-theoretical nature, dates back to the earliest attempts to deal with cellular automata from a mathematical (rather than merely phenomenological) viewpoint [28,29]. The diagram's proportions remind one that surjective systems represent a vanishing fraction of the entire class, or a *subset of measure zero*—in the sense that, as one increases a cellular automaton's complexity in terms of size of the state alphabet and of the neighborhood, the fraction of cellular automata that are surjective goes to zero; similarly for injective systems.

It turns out that there are no cellular automata that are injective but not surjective [30]; we indicate this by graying out the corresponding area, which henceforth will no longer be of concern to us. Thus, injective CA make up a vanishing subset not only of the whole set but also of even just the surjective ones.

We shall denote by $\mathcal{C}^+$ the minuscule class of systems that are *both* surjective and injective, called *invertible* (or, in physics, "microscopically reversible"). As we mentioned in Section 2 and shall more amply discuss relate in Section 5, any invertible cellular automaton – at least up to two dimensions – can indeed be rewritten as a lattice gas; thus, at least up to 2-D, all of $\mathcal{C}^+$ is in $\mathcal{L}$.

In Section 6 we discuss the class $\mathcal{C}^\circ$ of *almost invertible* cellular automata – those that are surjective but not injective – and we prove that, for any number of dimensions, *none* of $\mathcal{C}^\circ$ is in $\mathcal{L}$.

What is left is the class $\mathcal{C}^-$ of *locally lossy* cellular automata—that is, those that, not being especially marked as invertible or surjective, are (as one used to say) in "general position" (we'll give a more concrete characterization of them in a moment). These, which make up the bulk of all cellular automata, are this paper's main concern. Only a few are known to be presentable as lattice gases. Here we show, by means of an explicit construction, that all one-dimensional cellular automata of this class are so presentable, and indicate how this construction might be extended to multidimensional systems. On the basis of this evidence, we conjecture that *any* cellular automaton, with the exception of those of class $\mathcal{C}^\circ$, can be rewritten as a (possibly much more complicated) lattice gas; in other words, that $\mathcal{L} = \mathcal{C}^+ + \mathcal{C}^-$ (cf. diagram (1)).

In brief, concerning what parts of $\mathcal{C}$ are in $\mathcal{L}$,

- $\mathcal{C}^+$ is "in" (proved by Kari [18] up to 2-D).
- $\mathcal{C}^\circ$ is "out" (proved here).
- $\mathcal{C}^-$ is "in" (proved here for 1-D; conjectured here for the general case, with an actual working 2-D example; the latter provides a plausible outline for a general proof).

4. Local recognition criteria for $\mathcal{C}^+$ and $\mathcal{C}^-$

The categorical characterization of an invertible CA according to diagram (1) – that it be surjective and injective, that is to say, that every global state (or configuration) must have one and only one predecessor – is rather impractical, since configurations are countably infinite objects and their number is uncountably infinite.

In [36] we had provided a practical *local* criterion for recognizing invertibility—one that involved only local maps applied to a finite number of sites; namely,

Lemma 1. *There is an effective procedure for deciding, for any two local maps f and f', whether the corresponding global maps F and F' are inverses of one another.*

In a similar vein, our term "locally lossy" for a noninjective, nonsurjective CA reflects the following practical local criterion (Lemma 2), which is the only one that we'll have a need for in this paper, and which likewise involves only a CA's local map applied to a finite number of sites.

In both cases, we feel that the reader may gain more immediate access to the arguments and constructions discussed below by taking the local recognition criterion as the very *definition* of the kind of system in question, and the categorical properties as an equivalent, more elegant but more remote, characterization.

We recall that the *neighbors* of a site i are those sites from which the local map f takes its arguments at time t in order to compute i's new value for time $t + 1$. The *neighborhood*[4] X is a function which, when applied to a site i, yields the site's neighbors $i + j_0, \ldots, i + j_h$; it can thus be thought of as the collection $j_0, \ldots, j_h$ of these offsets. (In d dimensions, sites i and offsets j are represented by d-tuples.)

We shall call *patch* any collection of sites, and *pattern* the collective state of a patch. If I is a patch, $X(I)$ is defined in the obvious way as the *union* of the neighborhoods $X(i)$ for all sites i of the patch. When X is understood we write $\bar{I}$ for $X(I)$ and call it the *(causal) closure* of I—as it consists of all those sites the knowledge of whose state may be needed[5] in order to determine the new state of I. Just as f maps the state of the neighborhood $\bar{i}$ of a site i to that of the site itself, so does the induced function $f^{(I)} : A^{\bar{I}} \to A^I$, which is but a spatial iterate (a convolution) of f, map the state of a patch's I closure, $\bar{I}$, to the state of the patch itself. When the patch I is understood, there will be no need to distinguish between f and $f^{(I)}$. The *global map* F of a cellular automaton of local map f is $f^{(\mathbb{Z}^d)}$, where $\mathbb{Z}^d$ represents the entire infinite d-dimensional array of sites. If p is a pattern on a patch I, then I is the *support* of that pattern. A pattern P that has no predecessors, i.e., for which the equation $f(P') = P$ has no solutions, is called an *orphan* pattern.

We define a cellular automaton *locally lossy* if it has finite orphan patterns, that is, if for some finite set of sites I the application of f to $\bar{I}$ fails to yield all possible states for I itself.

Lemma 2. *A cellular automaton is locally lossy iff it is not surjective.*

Proof. If in a CA there is a finite patch I that has an orphan pattern, then any configuration containing this pattern is itself an orphan—and thus the CA is not surjective.

To prove the converse, i.e., that the CA is surjective if there are no finite orphan patterns, assume that the CA is *not* surjective, and thus has a configuration C with no predecessors under the global map F. Take a sequence $C_0, C_1, \ldots$ of finite patterns each containing the preceding one, all agreeing with C on their respective supports, and such that these supports cover the whole space of sites. If there are no finite orphans, each pattern C_j will have, on the closure of its support, at least one predecessor that we'll call B_j. If we extend in any manner B_j to the entire space we get a configuration, K_j. The sequence $K_0, K_1, \ldots$ being defined on a compact space,[6] has at least one accumulation point K, and from it one can extract a subsequence $K_{j_0}, K_{j_1}, \ldots$ that has K as its limit. By construction, $F(K_{j_n})$ coincides with C on the support of C_{j_n}; by continuity, $F(K) = C$, which contradict the *absurdum* hypothesis that C had no predecessors. (This proof is patterned after one by Fiorenzi [12, Prop. 3.2.7].) ■

See footnote 9 for ties to other equivalent characterizations.

We use *decidable* as a synonym for 'recursive', and *semidecidable* for 'recursively enumerable'. A consequence of Lemma 1 is that the class $\mathcal{C}^+$ is semidecidable. Similarly, Lemma 2 implies that the class $\mathcal{C}^-$ as well is semidecidable. From this, it follows that the class $\mathcal{C}^\circ$ itself must be not only undecidable, but also *not even semidecidable*. In fact, if $\mathcal{C}^\circ$ were semidecidable, all three classes $\mathcal{C}^+$, $\mathcal{C}^\circ$, and $\mathcal{C}^-$, which make up a partition of the universe C, would be *decidable*. But $\mathcal{C}^+$ had been proven *undecidable* by Kari [17], which leads to a contradiction!

We shall call *regular* a cellular automaton for which either invertibility or noninvertibility is eventually recognizable by local means (cf. Section 4), and thus one of class $\mathcal{C}^+$ or $\mathcal{C}^-$, and *singular* one that is not recognizable in that way, and thus of class $\mathcal{C}^\circ$. The results of this paper will make this terminology particularly suggestive (see Proposition 2).

5. Invertible cellular automata

The state of the art in invertible cellular automata was first reviewed in [36]; to avoid much repetition, we assume some familiarity with that material and with Kari's recent follow-up [20]. In the first of those two papers, one of us conjectured that all invertible cellular automata can be rewritten as lattice gases. This conjecture was subsequently proved, by Kari [18] for up to two dimensions[7]; this in spite of the undecidability of invertibility itself.[8] This undecidability nevertheless extracted a steep price; namely, there can be no computable upper bound to the increase in complexity (in terms of size of the state alphabet and of the neighborhood) in going from a cellular automaton to a functionally equivalent lattice gas. What this means in practice is that, though an invertible cellular automaton can always be emulated by a lattice gas, the latter may need arbitrarily more complex machinery per site.

A strictly analogous undecidability issue applies to the present quest for lattice-gas counterparts of locally lossy cellular automata, and consequently we must expect analogous complexity tradeoffs.

---

[4] Also called *neighborhood function*, *neighborhood index*, or *neighborhood template*.

[5] At least for *some* f having that neighborhood.

[6] The configurations set of a CA with the natural product topology is *compact* by Tychonoff's theorem. Moreover, the global map of a CA is by construction *continuous* with respect to this topology.

[7] A later paper by Durand–Lose [8] addressed the conjecture in the general case, and actually gave a construction for any number of dimensions. That construction, however, only provides an *embedding* rather than an isomorphism; in other words, it makes use of *ancillary variables* (see the considerations preceding Theorem 1). Though the resulting system is indeed an invertible LG, its state set is augmented with respect to that of the original CA; the ancillary variables themselves must be given specific initial values to guarantee correct behavior.

[8] The invertibility of a dynamical system on the basis of its presentation as a cellular automaton is undecidable—though semidecidable [36].

6. Almost invertible cellular automata

Here we use, for the class $\mathcal{C}^\circ$ of surjective but not invertible cellular automata, the mnemonic *almost invertible*. In fact, invertible cellular automata (class $\mathcal{C}^+$) do not lose information at all, while locally lossy cellular automata (class $\mathcal{C}^-$) lose in one step a nonzero amount of information *per site*. On the other hand, the automata of class $\mathcal{C}^\circ$, while *not* information-lossless, lose in one step such a small amount of information over the entire *infinite* array of sites that the loss *per site* is zero just as in the invertible case.[9]

The simplest example is obtained by replacing the AND gates of Fig. 1 with XOR gates. In that case every infinite configuration has exactly *two* predecessors, as in

$$\cdots 000001111 \cdots \searrow$$
$$\quad\quad\quad\quad\quad\quad\quad \cdots 000010000 \cdots$$
$$\cdots 111110000 \cdots \nearrow$$

and thus the dynamics loses just *one bit* of information per time step *over the entire array*.

Our question in this section is what part of $\mathcal{C}^\circ$ is in $\mathcal{L}$, that is, which almost invertible cellular automata can be rewritten as lattice gases.

For the purposes of mathematical analysis, a chief advantage of a lattice-gas presentation of a dynamical system over a cellular-automaton presentation is that, with the former, the system directly inherits certain categorical properties that can be ascertained, by mere inspection, on the presentation itself. Namely,

Lemma 3. *If a shift dynamical system admits of a lattice-gas presentation, then it is injective (resp. surjective) if and only if every node of the lattice gas is.*

Proof. First let us observe that if S is a Cartesian product of finite sets and F is a componentwise map on S, that is, $S = \prod_{i \in I} S_i$ and $(F(s))_i = f_i(s_i)$, then F is surjective iff each of the f_i is surjective. In fact, if $f_i(S_i) = S_i$ for all i, then $F(S) = S$; if, on the contrary, $x_i \in S_i \setminus f_i(S_i)$ and $(s)_i = x_i$, then $s \in S \setminus F(S)$. This is also true for injectivity: there can be two s's with the same image under F if and only if there are two s_i's with the same image under f_i for some i.

We've seen in Section 1 that, in a lattice gas, the state alphabet A is the Cartesian product of n factor alphabets, $A_1 \times \cdots \times A_n$. In turn, the state set for the entire system is $= A^{\mathbb{Z}^d}$, where d is the number of dimensions of the site array. Using Fig. 2, where $d = 1$ and $n = 2$, for sake of illustration, we can write S as

$$S = \cdots \times \underbrace{A_1 \times A_2}_{x-1} \times \underbrace{A_1 \times A_2}_{x} \times \underbrace{A_1 \times A_2}_{x+1} \times \cdots$$

where x is the site index. Even though it is true that the local map g maps $A_1 \times A_2$ into $A_1 \times A_2$, one may observe that, while output 1 of site x is used as input 1 of the *same* site, output 2 of site x is used as the input 2 of the *preceding* site, $x-1$, and so forth. In general, the signal transport operation denoted by the arrows performs a reshuffling of homologous state components between neighboring sites. This local coordinate reshuffling, from which there ensues at the global level a mere state reshuffling within S, is obviously invertible, and thus immaterial for the sake of injectivity and surjectivity. The thesis follows from the previous observation. ∎

Now, for a *finite* set – like the state alphabet A of a lattice gas – it is evident by the pigeonhole principle that a transformation f on it cannot fail to be injective without failing to be surjective as well, and vice versa; that is, if there is a point of A that is not in the image of f, then there must be some other point that has two counterimages. Thus a lattice gas can only be either both injective and surjective — that is, invertible – or neither – that is, locally lossy. We thus have that $\mathcal{L} \subset \mathcal{C}^+ \cup \mathcal{C}^-$, or

Theorem 2. *No almost invertible cellular automaton can be rewritten as a lattice gas.*

7. Locally lossy cellular automata

All that is left at this point is determine which locally lossy cellular automata (these, as we've remarked, comprise *virtually all* cellular automata) can be rewritten as lattice gases. There is little in the literature to guide one in this task.

7.1. Second-order cellular automata

Indeed the only systematic result concerns *second-order* cellular automata, where the new state of a site x at time $t+1$ depends not only on the state at time t of a number of neighboring sites but also on the state at time $t-1$ of x itself, as in Fig. 4. Of course, these can be rewritten as ordinary (i.e., first-order) cellular automata by means of a transformation analogous to the Legendre transform of classical mechanics: the second-order "Lagrangian" system with state set Q (the solid lines
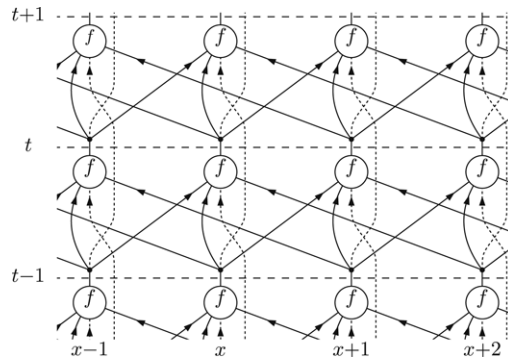
Fig. 4. A second-order cellular automaton. This example is in one dimension and has three neighbors per site ("left", "center", and "right of right"), besides an input from the previous time, indicated by a dotted line. Alternatively, the dotted signal can be thought of as a second state-component, and the system viewed as a first-order one with two coupled equations, like a Hamiltonian system.
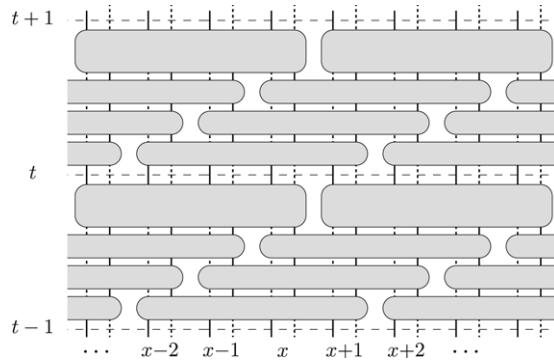

Fig. 5. Overall structure of a lattice gas representing a system isomorphic to that of Fig. 4. Note that each lattice gas node straddles *four* sites, and that it takes four layers of nodes to process one time step.

traversing time t in the figure) is viewed as a first-order "Hamiltonian" system with state set $P \times Q$ (the Q component is denoted by the dotted lines as they traverse time t).

In our 2004 paper [37] we showed that all second-order cellular automata, invertible or not, can be rewritten as lattice gases. For example, the cellular automaton of Fig. 4 can be rewritten as a lattice gas having the format of Fig. 5. There, each of the shaded blocks is a very simple 8-input, 8-output logic function (most signals happen to go through it unchanged), and the overall complexity of the machinery per site and time step is not much larger than that of the original cellular automaton. Note that each lattice-gas node straddles *four* of the original cellular-automaton sites, and it takes four layers of nodes to process one time step. Thus, the *structure* of the LG repeats itself only every *four* sites and every four layers. However, the *function* computed by the lattice gas is translation-invariant for shifts of a single site, just as the original cellular automaton. That is, we achieve a dynamical law having a given spatial regularity in spite of using a mechanism whose regularity pitch is four times as coarse!

The straddling of several sites and the attendant coarsening of the structure's translation group is a general feature of lattice-gas presentations vis-à-vis cellular automata presentations—it is, as it were, the price one has to pay to be able to sense many neighbors without making recourse to signal fanout. In the case of second-order cellular automata, the pitch ratio (4, in the present example) coincides with the span of the neighborhood[10]; in general, though (and, specifically, in the case of invertible cellular automata), the pitch ratio is an unbounded function of the cellular automaton's complexity, owing to the undecidability mentioned in Section 5. Thus one must expect to find cellular automata for which the simplest lattice-gas counterpart, if one exists, has an arbitrarily larger and more complex "unit cell" (cf. [38]) than the original cellular automaton. We shall see that this is also the case with locally lossy cellular automata, and touch upon the concrete reason why.

---

[9] This is just a more intuitive way of stating Moore and Myhill's classic theorem [28,29], namely, that (in Maruoka and Kimura's terms [27]), *a cellular automaton's global map is surjective if it admits of no erasable patterns*. See Calude's section "Randomness in cellular automata" [6, 9.5] for a review and consolidation of these concepts.

[10] In more than one dimension, this "ratio" will be a *vector* with one component for each dimension.
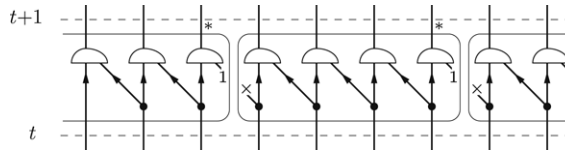
Fig. 6. Here the cellular automaton of Fig. 1 has been sliced into four-site segments, and each segment lumped into a lattice-gas node.
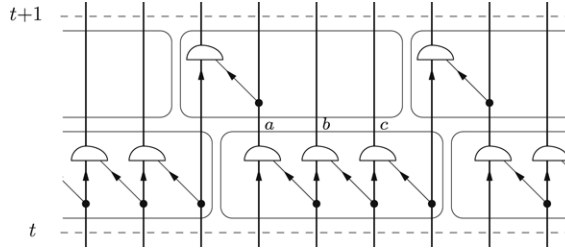

Fig. 7. Adding a second layer of nodes to bridge the gaps left from the first layer (cf. Fig. 6). Unfortunately, the lone AND gate on the second layer gets its right-neighbor signal as it will be at time $t + 1$, rather than as it was at time t.

7.2. When information loss helps

We shall now sketch in an intuitive way the argument on which our proof and conjecture rest. Let us go back to the "AND" cellular automaton of Section 3 and attempt to transform it into a lattice gas. A naive approach would be to cut it up into segments of, say, four sites each, and turn each segment into a lattice-gas node, as in Fig. 6. In this way, the neighbor links that used to connect adjoining segments are severed[11]: neighbor data coming out of the left of each segment are lost, while the right input to the rightmost gate of each segment remains unspecified. We may force this input to a constant value—say, 1.

We have now obtained a lattice gas with 4-input, 4-output nodes each straddling four of the cellular automaton sites. The picture inside each node – the AND gate and the fanout junction – is merely a reminder of what *function* the node is supposed to compute, not a representation of its internal structure. In fact a lattice-gas node, like any logic gate, has no internal structure at all associated with it—only an assigned correspondence (one may think of it as a lookup table) between input and output states. Specifically, *no actual fanout* (in some presumed internal mechanism?) is implied by the use of fanout icons in the picture.

Moreover, the data transformation performed by this lattice gas from time t to $t + 1$ is clearly identical to that performed by the original cellular automaton—*except* for each node's rightmost output (marked with a star), which just echoes the input from the corresponding site without AND ing it with its right neighbor. Our next task will be to fix this problem without giving up the lattice-gas format.

In order to avoid "lateral" transmission of information within the same layer (which is against the rules for an LG—and indeed it would create a path for instantaneous transmission of information across infinite distances) we may try to add a second layer to the lattice gas, with the nodes of this layer bridging the gaps left from the first layer, as in Fig. 7. In this way, the lone AND gate on the second layer does get, as intended, a right-neighbor signal as well as a center signal. However, while the center signal is that of time t, as desired, the right-neighbor signal has already been processed by the first layer and thus has the state it will display at time $t + 1$. We do get an LG, it is true, but still one with the wrong dynamics!

To obviate the above problem, we could get the right-input to the second-layer gate by tapping line a ahead of first-layer processing, at a', so as to forward a copy of this as-yet-unprocessed signal (dotted line) directly to the second layer, as in Fig. 8. In this way we certainly obtain the right dynamics – that of Fig. 1 – between t and $t+1$, but the intervening mechanism is no longer a lattice gas! In fact, the first-layer nodes have four binary input lines but five output lines – and vice versa for the second-layer nodes – while we had postulated (in Section 1) that lattice-gas nodes should have *equal* input and output state sets. Intuitively, to perform a transformation involving four state variables, we have taken the liberty here of introducing *ex nihilo* a fifth state variable, for intermediate storage. From a physical viewpoint, the dynamics relies on external degrees of freedom and is no longer self-contained.

Fortunately, there is one feature that hasn't yet been brought into play; namely, that we are dealing with the class of *nonsurjective* cellular automata, which lose information on a local basis—that is, a nonzero amount per site. Specifically, even as the input lines at time t range over all possible combinations of binary states, not all eight possible states for lines a, b, c will actually be produced; in fact, one may verify that, with the given dynamics, state 101 will *never* appear at the output. In other words, in going from time t to $t + 1$ with the AND rule, the 3-bit "channel" $\langle a, b, c \rangle$ is not utilized at full

---

[11] By definition, a distributed dynamical system cannot have "lateral" transmission of information – that is, between nodes that belong to the *same* time slice – because that would introduce infinite regress ("infinite speed of propagation of information") and thus leave the global state under- or over-determined.
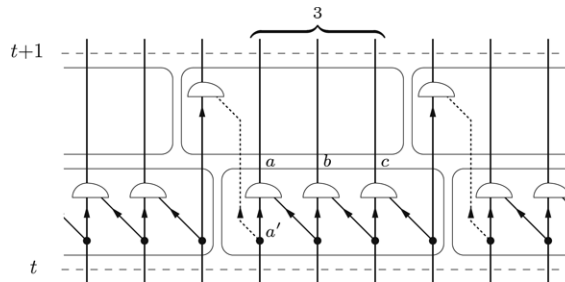
Fig. 8. Here we tap the state of the a line at a' – *ahead* of first-layer processing – and forward it (dotted line) directly to the second layer.
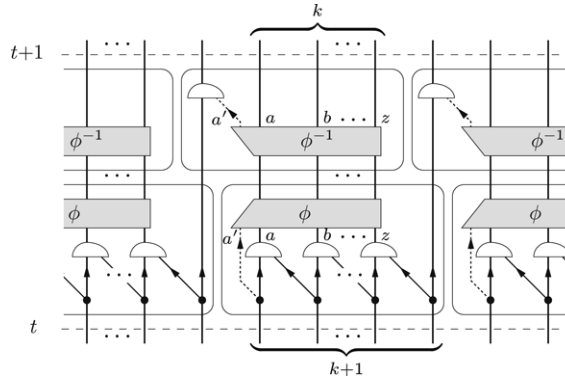

Fig. 9. To avoid running an additional signal line between first and second layer, in the first layer we use an encoder ϕ to squeeze the dotted signal into the underutilized channel provided by the next k sites, and in the second layer a decoder ϕ^{-1} to get this signal back out.

capacity. Can one use the spare capacity to transport some unrelated information? For instance, that now carried by the dotted line in Fig. 8?

To this purpose, let us set up a two-layer lattice gas just as in Fig. 8, but with a k-site wide channel, with k to be chosen large enough to provided the required capacity. As shown in Fig. 9, in the first layer a $(k+1)$-to-k encoder ϕ will squeeze the extra signal (dotted line) together with the k underutilized signal lines into k better utilized lines; conversely, in the second layer a k-to-$(k+1)$ decoder ϕ^{-1} will separate the extra signal from the other k signals and deliver it to its destination gate. Such a coder/decoder pair is termed, in the trade, a *codec*.

What is the minimum codec width k for the given AND-rule CA? For a first rough estimate, let's observe that by taking one step of the AND rule starting from the maximally random ensemble U_0 of configurations, which has an entropy of one bit per site, we obtain an ensemble U_1 where at any site a 1 will appear with a probability of 1/4. The entropy per site η of this new ensemble will be no more than $1/4 \lg_2 4 + 3/4 \lg_2 4/3$, or $\approx .811$ (row 1 of Table 1)—and possibly somewhat less because of local correlations. In fact, a tabulation of η for blocks of increasing length n allows us to estimate an actual entropy density of approximately .7 bits/site, and thus an entropy defect of about .3 bits/site. This is the spare "channel capacity" per site of the AND CA thought of as an infinite array of binary lines filled with a noise distribution S_1.

This very crude level of analysis tells us that the width of the encoder may have to be of the order of $k = 4$—since with a .3 bit/site capacity it would take at least 4 sites to squeeze through one extra bit. A more precise analysis must take into consideration two additional factors:

1. We want a *finite*, *deterministic* encoder, which cannot tolerate errors and yet cannot, as in Shannon's theory, fix errors by sharing resources with adjacent blocks. In our case, a more appropriate quantity than the entropy H of a random variable is the *variety* V of a function. Intuitively, variety corresponds to the *number of distinct values* that the function takes over its domain. Like entropy, variety is customarily expressed on a *logarithmic* scale [2] in an appropriate base (we shall be using base 2 for the moment). The (logarithmic) variety $V(y)$ of the output k-tuple $y = \langle a, b, \dots, z \rangle$, when the $k + 1$ inputs to a block at time t range over all their possible 2^{k+1} values, is given by column V of Table 1. We see that only for $k = 8$ does the variety of the "new state" drop below 7 (the boldface value 6.83 on row 8), thus leaving enough space for one extra bit (for example, a') to be squeezed by the encoder into the same k lines.
2. On the other hand, a' is in general to some extent correlated with the y k-tuple, since a' is part of y's "cause". Thus we may expect that a' may contribute less that one bit of additional variety to the $(k + 1)$-tuple $\langle a', a, b, \dots, z \rangle$. In fact, it turns out that in our case the variety V' of the $(k + 1)$-tuple $\langle a', a, b, \dots, z \rangle$ drops to no more than k already for $k = 4$ (boldface value 4 for V' on row 4). That is the threshold we were looking for!

We conclude that, for the CA in question, an equivalent LG presentation with two layers of blocks of width $n = k + 1$ and an appropriate *codec* of width k is possible as soon as $k \geq 4$.

Table 1
Total entropy H, entropy density (per site) η, and variety per site V of the output pattern of width k produced in one step by the "AND" cellular automaton of Fig. 1 from a maximally-random input neighborhood of width $k + 1$

k	H	η	V	V'	
1	0.81	.811	1.00	1.58	2
2	1.55	.774	2.00	2.32	3
3	2.25	.750	2.81	3.17	4
4	2.95	.737	3.58	**4.00**	5
5	3.65	.730	4.39	4.81	6
6	4.35	.725	5.21	5.61	7
7	5.05	.721	6.02	6.43	8
8	5.75	.718	**6.83**	7.24	9
9	6.45	.716	7.64	8.05	10
10	7.15	.715	8.46	8.86	11
11	7.84	.713	9.27	9.67	12
12	8.54	.712	10.08	10.48	13
13	9.24	.711	10.89	11.30	14
14	9.94	.710	11.70	12.11	15
15	10.64	.709	12.51	12.92	16
16	11.34	.709	13.32	13.73	17

The last column, V', gives the variety when the output k-tuple is augmented by the input signal a' (see Fig. 9). Note that, because of internal correlations, this extra binary line only adds about 1/2 bit of variety.

8. The one-dimensional case in general

The foregoing construction was given for a simplest nontrivial CA (cf. Section 3), which (a) is one-dimensional, (b) uses a 2-state alphabet (the Boolean values 0 and 1), (c) has a 2-element neighborhood template (consisting of the two offsets 0 and 1), and (d) uses the AND function as a local map on this neighborhood.

Here, while remaining within one dimension, we shall generalize that construction to any *state alphabet A*, *neighborhood X*, and *local map f*. In subsequent sections we shall pursue the most general goal along these lines, i.e., a construction method that applies to any *number of dimensions*.

In one dimension, the *diameter* of the neighborhood is the distance between leftmost and rightmost neighbors, or $j_h - j_0$. (In the AND CA used in Section 7.2 the offsets were 0 and 1 and thus the neighborhood diameter was $h = 1 - 0 = 1$.) A neighborhood may be *sparse*, in the sense that some offsets between the lowest and the highest are missing; the diameter is not affected by sparseness. Henceforward, without loss of generality, we shall assume that the neighborhood is not sparse (gaps may always be filled by dummy neighbors), and shall add a constant to all offsets so as to make the least offset 0. With this convention, the neighborhood diameter coincides with the highest offset h, where $h + 1$ is the number of neighbors.

The construction template we propose below for the general one-dimensional case, illustrated by the sequence of steps of Fig. 10, has two adjustable parameters. The first is the *neighborhood diameter h*; the second, the *codec size k* (i.e., the number of sites spanned by it). The corresponding block size will be $n = k + h$.

In Fig. 10 we have used two conventions to avoid clutter. First of all, instead of having $h + 1$ neighbor wires converge onto a gate as inputs (as, for instance, in Fig. 4), we have retained a separate wire only for the neighbor of offset 0; all other h neighbors are picked up by a side extension of the gate itself which senses them but also lets them go through, so that they may be sensed in turn by all the gates of which they are neighbors. We thus have a notation suggestive of a cross-point matrix connecting vertical "source" wires to horizontal "destination" wires. This notation is standard in reversible logic [34] and quantum computing [4]. Secondly, instead of indicating an indeterminate number of repetitions of a certain feature (such as the circuitry of one site) by explicitly giving the first and the last occurrence of the feature and implying all the intermediate occurrences by ellipsis dots ("..."), as in Fig. 9, we imply the intermediate occurrences by a *single* grayed-out icon, as well as a brace indicating how many occurrences are intended in all. Thus, the h neighbors of a site (not including the site itself, or neighbor 0) are indicated by an icon for neighbor 1, one for the last neighbor h, and a single grayed-out icon standing for the remaining $h - 2$ neighbors.

For the construction to work as intended, the k lines running from the encoder ϕ in the first layer to the decoder ϕ^{-1} in the second must be able to carry, distributed among themselves, not only the corresponding k new states just computed within the first layer, but also information about the *old* state of the leftmost h of those k sites, which is still needed for updating h left-over sites in the second layer. The encoder must be wide enough for this purpose. To show that a k large enough to provide the required extra capacity can always be found we shall be making use of *Fekete's lemma* [11], which we restate here in a form similar to that used by van Lint and Wilson [39].[12]

Lemma 4 (*Fekete*). *Let $g : \mathbb{N} \to \mathbb{R}^+$ be a function for which $g(i + j) \leq g(i)g(j)$ for all $i, j \in \mathbb{N}$. Then $\lim_{i \to \infty} g(i)^{1/i}$ exists and equals $\inf_{i \geq 1} g(i)^{1/i}$.*

---

[12] As remarked by a referee, if one is not only told that the CA is locally lossy but is also presented with a specific orphan pattern, then, instead of the existential proof represented by Fekete's lemma one can directly home in on a bound by using an argument analogous to Moore's [28].
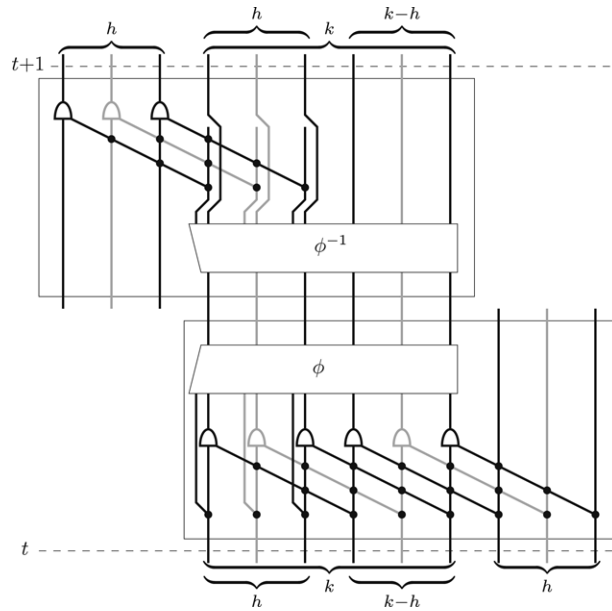
Fig. 10. General template for converting into a two-layer lattice gas an arbitrary one-dimensional locally lossy cellular automaton. The dome-like "gate" denotes the local map f. The parameter h is the neighborhood diameter; k is the size of the codec, and $n = k + h$ the block size. Note how, in the second layer, neighbor inputs other than the 0 neighbor are not taken from the new-state lines but from the corresponding old-state lines, which had been store-and-forwarded from the first layer. Notation conventions are explained below.

Theorem 3. *Any one-dimensional locally lossy cellular automaton can be rewritten as an isomorphic lattice gas.*

Proof. Consider a one-dimensional CA of state alphabet A of cardinality a, neighborhood X of diameter h, and local map $f : A^X \to A$.

Given a patch I and its closure $\bar{I}$, for any state $x \in A^{\bar{I}}$ the local map f will determine the corresponding new state $y = f(x)$ (with $y \in A^I$). Observe that, if the CA is not trivial, at least one state of I must have more than one pre-image among the states of $\bar{I}$. On the other hand, there may be states of I that do not arise from *any* state of $\bar{I}$; these are I's orphans. By definition, a CA is locally lossy if for some i the corresponding patch I has orphans.

Let us denote by $g(i)$ the variety of the range of f, that is, the number of distinct states for I that actually arise from $y = f(x)$ as x runs over f's domain, $A^{\bar{I}}$. Consider now two adjacent patches, I and J, spanning, respectively, i and j sites, and their juxtaposition IJ of length $i + j$. If the I portion of a pattern on IJ happens to be an orphan of I, then *that entire pattern* is an orphan for IJ, immaterially of whether or not the J portion of it is an orphan for J. It immediately follows that $g(i+j)$ can be *at most* as large as $g(i)g(j)$, and Fekete's lemma as given above applies verbatim.

If we now express variety in a logarithmic fashion (see Section 7.2) using a for a base – that is, if variety is represented by the quantity $v(i) = \log_a g(i)$ – then the appropriate form of Fekete's lemma to be applied to this quantity is the sub-additive (rather than sub-multiplicative) one, that is,

$$v(i + j) \leq v(i) + v(j) \implies \exists_{\eta \in \mathbb{R}} \lim_{i \to \infty} v(i)/i = \inf_{i > 1} v(i) = \eta.$$

Since the variety $v(i)$ is in any case comprised between i (when the range of f is all of a^i) and 0 (when all of I's neighborhood states map into one and the same state for I), the above limit η is a number between 0 and 1.

For the construction of Fig. 10 to work, the k lines of the codec (which collectively have a variety capacity of k) must be able to distinguish between all effectively occurring combinations of values for the new state of k sites together with the old state of the adjacent h neighboring sites. Therefore, the joint variety $\tilde{k}$ of all these combinations must not exceed the variety capacity k of the codec "channel". An upper bound to $\tilde{k}$ is given by the sum of the marginal varieties, namely, h for the h neighbors and k' (to be determined below) for the new state of the k sites.

If the CA is locally lossy, then $v(i)$ is eventually less than i, and thus $\eta < 1$. In this case, given any desired variety h, and $\epsilon > 0$ such that $\eta' = \eta + \epsilon < 1$, for large enough k we have $v(k)/k < \eta'$ and $k - v(k) > k(1 - \eta') \geq h$. In other words, for any neighborhood diameter h we can always find a patch K of length k having a *variety defect* $k - v(k)$ at least as large as h. A codec built on this patch will be able to transmit through k lines the variety of the new state of k sites together with that of the old state of h of these sites. $\blacksquare$

We can apply Theorem 3 to Wolfram's so-called "rule 110" [42] and conclude that

Corollary 1. *There exist noninvertible computation-universal cellular automata that can be isomorphically rewritten as lattice gases.*

Proof. It is easy to verify that a codec of sufficient capacity for the construction of Fig. 10 to work – with $h = 2$ and rule 110 as a local map – can be found as soon as $k \geq 17$. ∎

Remark 1. In [19], Kari determined that, for rewriting the minimum number of layers necessary

9. A two-dimensional example

Before attempting to extend the above result to an arbitrary number of dimensions, it will be expedient to verify at the very least that the one-dimensional construction of Section 7.2, for a 2-input AND cellular automaton, can be leveraged (with some creativity) to the analogous two-dimensional case, that is, a (2×2)-input AND automaton.

In the 1-D case of Section 7.2, our CA-to-LG construction yielded a 2-layer lattice gas that employed one codec to negotiate the information transfer between layers 1 and 2. The codec was four sites wide, and in either layer the block spanned by the LG node was five site wide—the 1 extra site corresponding to the neighborhood diameter, which was, in that case, 1.

In the present 2-D case, five layers (instead of merely two) will be required for the state of the lattice gas to advance through one time step of the cellular automaton. Four different codecs will be needed to negotiate the information transfer between adjacent layers, i.e., $1 \to 2, 2 \to 3, 3 \to 4$ and $4 \to 5$. As for the size of the LG blocks, which is determined by the size of the smallest codecs that will do the job, let us remark that the most critical passage is from layer 1 to layer 2 (as we shall see, much less "spare capacity" is needed at the next three passages). Using the lossiest (the "most noninvertible") nontrivial CA rule on a 2×2 neighborhood, namely the AND of the four neighbors,[13] a codec of size 4×4 will have enough capacity to pass the required edge-neighbor information from layer 1 to layer 2.[14] Therefore, since the local map is of the format $2 \times 2 \to 1 \times 1$, and thus has a neighborhood diameter of 1 in either dimension, to host a codec of size 4×4 we shall need an LG block size of 5×5 (cf. the analog situation in the 1-D case, as in Fig. 9).

In Fig. 11 we describe the function computed by each layer in terms of a sequence of logical steps (4 for the first layer, 2 for the last, and 5 for each of the intervening three). We stress again that such a sequence gives a *heuristics* for identifying that function but is not meant to be a description of an *internal mechanism* for it. Each layer will consist of LG nodes each simply taking its inputs from a block of 5×5 sites and returning its 5×5 outputs to a block of the same size; there is no *logical* need to analyze this function as a "computation", i.e., as a series/parallel composition of simpler functions. What's more, in a *physical* implementation of the function, in order to have the maximum thermodynamic efficiency (see Section 12) it may be essential to make use of a custom physical interaction that makes the 5×5 signals interact with one another *all at once*! Whatever the heuristics, in the end it may be more appropriate to visualize each node merely as an indivisible interaction—a mere lookup table.

Layer 1 (b, c, d)

In Fig. 11, which may be viewed literally as a "flipbook", we enter layer 1 immediately after $t = 0$; a dot indicates that a site is in the "old" ($t = 0$) state. At step 1b the sites are grouped into 5×5 blocks. At 1c, we compute the "new" state (the one that will be appropriate for $t = 1$), indicated by a circle, of all the sites that can view their entire 2×2 neighborhood *within that block*; a dot within a circle means that we are at present in possession of the old as well as the new state for a given site.

Next (1d) we ask ourselves which of the old states will still be needed for eventually computing the new state of the sites which we haven't been able to update yet; the 3×3 old states which have exhausted their function may now be dropped from consideration. At 1e we recognize that, owing to the local noninvertibility of the AND rule, the variety of the array of $4 \times 4 = 16$ new states is less than 16 bits. Therefore, the 16 lines corresponding to those sites have enough "channel capacity" to carry some extra information, specifically, the 7 old states which we are still carrying over. We thus construct, in analogy with what we did in Fig. 9, an encoder (gray area) that compresses those $16 + 7$ binary lines into just 16 lines which are passed on to the next layer. The 9 not-yet-updated sites that make the unshaded border are passed on verbatim to the next stage.

Layer 2 (a, b, c, d, e)

We now shift the block boundaries one site leftwards. At 2a the 16 encoded lines are picked up by a matching decoder which expands them into the original $16 + 7$ lines (step 2b). At 2c we update as many new additional sites as possible on the basis of old states that have just been made accessible within the block by shifting its boundaries; and at 2d we discard the old states for which we have no further need. Finally (step 2e) we prepare an encoder for squeezing through to the next level the 5 old states that still accompany their new states. Note that now we have more channel capacity available (deriving from a 5×4 rather than a 4×4 array of new states) for fewer lines to squeeze through (5 instead of 7); in other words, things become easier once cleared the information bottleneck of the first layer.

---

[13] This rule maps the 16 neighborhood states into 15 zeros and 1 one, while any surjective rule (and thus any invertible one) would need to map the 16 states into 8 zeros and 8 ones—see the "balanced map" criterion in [27].

[14] With 16 binary lines, this 4×4 codec has a capacity for $2^{16} = 65,536$ different messages, which is more than enough for the 52,886 different messages (combinations of new and old site values, as explained in point 2 near the end of Section 7.2) which we need to send through the codec from layer 1 to layer 2.
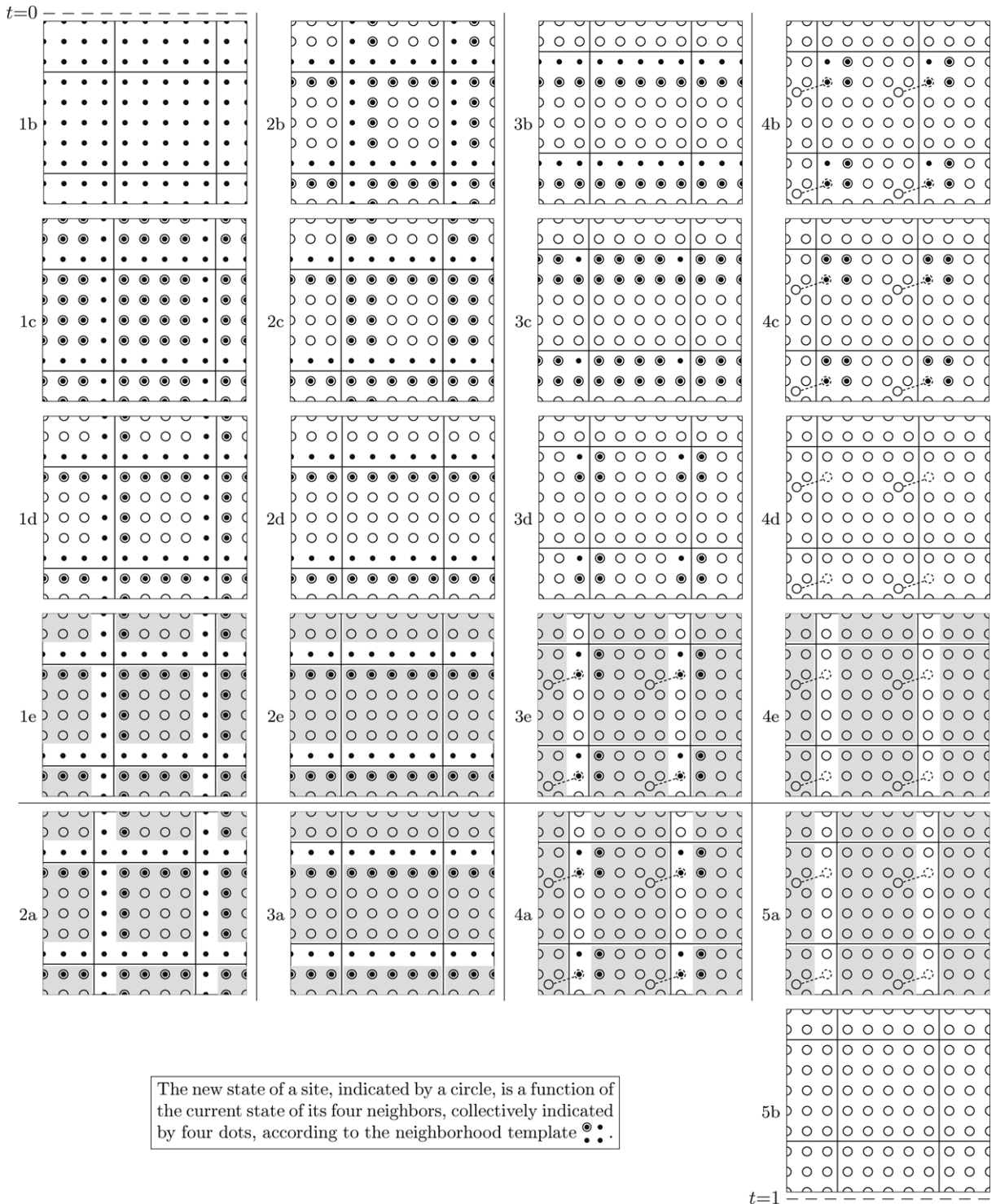
Fig. 11. Construction of a 5-layer lattice gas for the 2 × 2 AND cellular automaton.

Layer 3 (a, b, c, d, e)

Block boundaries are now shifted one site upwards. Layer 3 proceeds much like layer 2, with just one novelty. At the moment (step 3e) of packing 20 new states plus 2 old states into a 4 × 5 encoder, we realize that one site outside of the nominal span of the encoder has both old and new state with it, while its one line, to be carried over verbatim to the next layer, will have no room for both. What we do then is let the *old* state of that site go through in place, but "tuck away" into the encoder the *new* state, as indicated by the dashed line.

Layer 4 (a, b, c, d, e)

Block boundaries are shifted one site leftwards. Everything proceeds like in layers 2 and 3.

Layer 5 (a, b)

We shift block boundaries rightwards, i.e., back were they were at the end of layer 3. We unpack everything (step 5a) from the decoder, including the new state that had been tucked away at step 3e, and at 5b we return this state to its original site. Now ($t = 1$) that there are no more old states left we are ready to rechristen the new states 'old states' and start a new 5-layer cycle.

As long as the neighborhood remains 2×2, it is clear that the above construction pattern will work for a larger state alphabet and an arbitrary local map,[15] just as in Section 8; all that will be required is a large enough $k \times k$ block size.

In the 1-D case our construction of Section 7.2 yielded a 2-layer LG; in the 2-D case of Section 9, a 5-layer one. What can one say about the minimum number of layers needed in general? In [19], Kari showed that in d dimensions the rewriting of *invertible* CA into isomorphic LG, when possible, need not use more than $d + 1$ layers. If an analogous bound held for the *locally lossy* CA dealt with here, then the 2-D construction of Fig. 11 could be replaced by one with only 3 layers. Even if it were possible, we did not attempt such a compression, which may entail larger or more complex blocks in each layer; we feel that the more relaxed approach of Fig. 11 makes it easier to follow the construction's rationale.

10. General case

The 2-D construction of Section 9 can apparently be generalized to an arbitrary neighborhood size; that would make our 2-D result as general as the 1-D one. One may then wonder whether an analogous result holds for *any* number of dimensions. In support of that thesis, we'd like to observe that, as long as one retains the construction strategy implied by Sections 8 and 9,

1. The strategy does not depend on the size a of the state alphabet; the latter only influences the parameter k—the size of the codec.
2. The strategy does not depend on the diameter h of the neighborhood; the latter only determines the thickness of the "boundary layer" of neighbors whose old state has to be carried through until no longer needed.
3. The strategy does not depend on the specific local map f; the latter, assumed to be locally lossy, only influences, through the extent of the information losses (gaged by the Fekete's lemma's limit η of Theorem 3 or its multidimensional counterpart as in Lemma 5), the size of the codec.
4. The existence of codecs of appropriate sizes for the different stages is guaranteed by the multivariate version of Fekete's lemma (below).

Lemma 5 (*Multivariate Fekete's Lemma; Capobianco [7]*). *Let $f : \mathbb{N}^d \to \mathbb{R}^+$ be a function of d variables, subadditive in each, i.e., for any j*

$$f(x_1, \ldots, x_j + y_j, \ldots, x_d) \leq f(x_1, \ldots, x_j, \ldots, x_d) + f(x_1, \ldots, y_j, \ldots, x_d).$$

Let $\eta = \inf_{x_1, \ldots, x_d \geq 1} f(x_1, \ldots, x_d)/x_1 \cdots x_d$. For any $\epsilon > 0$ there exist $k_1, \ldots, k_d$ such that, if $x_j > k_j$ for all j, then $f(x_1, \ldots, x_d)/x_1 \cdots x_d < \eta + \epsilon$.

Therefore, each of the stages, analogous to those of Section 9, that may be needed for the d-dimensional case, is feasible for a large enough block size $(n_1, n_2, \ldots, n_d)$. All that is missing at this point (but this may well turn out to be a tall order) is the guarantee that, for any d, an appropriate finite sequence of "progressive partial updating" layers may be constructed so as to ultimately achieve, in a finite number of steps, a complete updating of the CA.

A bit of common sense will remind one that

- Just by looking at the increase in logical and graphical construction complexity in going from 1-D to 2-D, one may want an automated *proof* – or even an automated *proof generator* – to tackle the complexity of the sequence of steps needed for more dimensions and organize it in a comprehensible way. For instance, (a) Will the number of layers be essentially proportional to the number of dimensions, and thus to the number of (hyper)faces of a block? (b) Will one also need extra layers to account for (hyper)edges, . . ., and so forth down to the 0-dimensional vertices?
- Numerical verification of representative test cases is invaluable in complicated proofs. ("Pure" mathematicians may argue otherwise, but we suspect that, far from shunning such verifications, they actually do them automatically and subconsciously.) However, the computational burden of such tests grows exponentially (and *extremely* fast if one takes block *side* as a parameter), to the point that, even for Conway's "Game of Life" [13] – a moderately lossy 2-state, 3×3-neighborhood cellular automaton – the effective construction of the codecs is a well-nigh intractable task!

---

[15] Though, of course, still one that yields a locally lossy CA.

- There are encouraging and discouraging precedents in this area. As we already mentioned at the beginning of Section 2, a related conjecture, which had taken six years to prove for one and two dimensions, took another six years and a very complex construction to prove in the general case. On the other hand, in another argument entailing, again, surjectivity and injectivity in cellular automata, the authors of [1] had thought that their techniques were "in principle extendable to arrays of higher dimension"; since, however, these techniques were "difficult to manage beyond dimension one", they expected that "generalizations of their results to higher dimensions" would "most likely require a different approach". In the end (that is, almost twenty years later), the conjecture turned out to be wrong and its object undecidable [17]!

The evidence given, together with more work in progress on actual multi-dimensional exhibits, is strongly suggestive but not complete. For the moment, we prefer to propose the general case only as a conjecture, namely,

Conjecture 1. *Any locally lossy cellular automaton can be rewritten as an isomorphic lattice gas.*

In any event, it is remarkable how, in spite of classes $\mathcal{C}^+$ and $\mathcal{C}^-$ being at opposite extremes in the categorical classification (1), the construction strategy utilized here has strong analogies (blocks, layers, store-and-forward of information with repeated exchange through block edges) with those used by Kari and by Durand–Lose. This in spite of the fact that the latter critically rely on a CA's being strictly *information lossless* while ours critically depends on their being *locally lossy*!

11. Some consequences of the conjecture

From the truth of Conjecture 1 would derive the following remarkable consequences.

Proposition 1. *All cellular automata, except the vanishing subset $\mathcal{C}^\circ$, can be rewritten as isomorphic lattice gases.*

Proposition 2. *Regular cellular automata* (see end of Section 4) *are exactly those which can be rewritten as isomorphic lattice gases; singular cellular automata, those which cannot.*

12. Thermodynamic considerations

Here we shall briefly touch on some thermodynamic aspects of the foregoing purely mathematical results. Though we could have phrased the following purely in terms of *entropy*, without ever having to embody it into "energy" and "heat", there will be no harm in using the more familiar, though less general, picture.

As mentioned right before Theorem 1, a concrete physical system patterned after the "schematics" of a cellular automaton will require a steady supply of power to operate and a thermal sink capable of steadily dissipating this power – *no matter whether or not the underlying abstract dynamical system is invertible* – to support the "document duplicating services" continually performed by the fanout nodes and the "surplus-document shredding" performed by the many-input, one-output logic gates.

Speaking here, for simplicity, as if our conjecture were true (if not, our arguments will apply at least to the cases we have proved), for regular cellular automata it is in principle possible to build an alternate implementation, in the form of a lattice gas that does not require a power supply to operate.[16]

For those cellular automata that are invertible, besides no power supply, no *thermal sink* will be needed either, since, with Kari's and Durand–Lose's constructions, large enough lattice-gas blocks allow one to make use of all relevant correlations and thus operate in an efficient, dissipationless fashion. The system will work analogously to a planetary system—a complex nonlinear system[17] that can ideally run forever on its own initial energy, without requiring an external power source.

On the other hand, a system like Conway's "Game of Life" is intrinsically noninvertible, and so to implement it within our world's physics one will have to complement it with an ancillary "entropy drain" that is not explicitly represented within the cellular-automaton model. The combined physical system (cellular automaton plus ancilla) is one that exhibits friction and gradually converts "high-grade" energy into heat. However, our construction shows that, much as in the invertible case, this conversion need not entail more than the system's *initial store* of energy—no continual power injection from outside is needed for the system's operation. Consequently, throughout the system's *infinite course* of evolution, no more entropy will have to be drained from it than the information originally contained in it at time $t = 0$, which is at most $\log a$ per site. This behavior is like that of an isolated inertial planetary system in which some of the interactions are affected by friction.

As a dissipative system (whether planetary or "Game of Life"-like) evolves and its trajectories gradually merge, the system becomes effectively closer and closer to being invertible[18] and the attendant heat production per time step decreases. The heat integral over the entire infinite trajectory from a given initial state can never exceed the energy contents of the

---

[16] More precisely, even if some form of friction should be unavoidable because of the nonideal behavior of the implementing mechanism, yet there is no theoretical lower bound set by thermodynamics to the amount of this friction, and thus to the amount of power that will have to be supplied to overcome it.

[17] Even capable of general-purpose computation, as shown in a stylized way by Margolus's "billiard-ball" model of computation [26].

[18] For instance, a Game-of-Life "semaphore" or a "glider", taken in isolation, may be viewed as obeying an invertible dynamics.

initial state. In this sense, the system's state simply "flows downhill" by itself, rapidly at first but increasingly more slowly, until it mostly settles into local equilibria or local attractor states, which are essentially nondissipative. If the system is to be computation-universal, then, for certain initial conditions, flickers of dissipative nontrivial computation may still occasionally occur, but ever more rarely and sparsely (cf. Dyson [9]).

The perhaps surprising, but not at all paradoxical, conclusion is that a complex CA-like dynamics can run *forever*, faithfully applying its local map at each site and each time step, fueled, as it were, just by the negentropy of the *input data*, i.e., its own initial state.

A physically plausible scheme for CA-like digital computation fueled by just its input data had been proposed by Lent, Tougaw, and Porod [22], ostensibly realized by simple interactions between chains of bistable one-electron quantum boxes. There was some debate at that time as to whether such a scheme could actually work as purported—though objections mostly reduced to "the outcome seems too good to be true". The specific interactions sketched at that time may well have been too weak to support the desired behavior; nevertheless, our results show that a scheme like theirs is in principle feasible, though requiring the deployment, for "primitive computing elements", of rather complex, ad hoc physical effects.

13. Conclusions and perspectives

We have presented and discussed the domain of validity of Proposition 2, (a) proving that it applies to all 1-D cellular automata and some 2-D ones, (b) giving substantial evidence in favor of its applying to *any* number of dimensions, and (c) outlining a plausible path to completing that evidence. In any event, just results (a) above are sufficient to conclude that

- There exist nontrivial *noninvertible* cellular automata that can be rewritten as isomorphic lattice gases. These include cellular automata that are computation-universal.
- Much as in the case of invertible cellular automata, the conversion of a noninvertible cellular automaton into an equivalent lattice gas entails using LG unit cells that span a possibly very large number of CA sites. The resulting LG, then, while displaying the same *functional* pitch as the original CA, must make use of a coarser *structural* pitch—and thus of a more complex local mechanism.

  Intuitively, for a given computing task, the thermodynamic efficiency of a lattice gas – which unlike a CA does not indulge in wasteful "photocopying" and "shredding" practices – can only be achieved at the cost of using LG nodes that encompass a sufficiently large number n of CA sites. Such a large node is able to act as a "clearing house", matching "offer" and "demand" for specific pieces of information between faraway CA sites, routing information to successive destinations as needed ("information recycling") and in this way eliminating duplication and waste.

In Section 3 we mentioned the use of finite-difference schemes for the numerical integration of partial differential equations. In many cases, a number of alternative schemes have been proposed for dealing with the *same* differential equation; of these, some may be *much more complex* than others. Why should one go to such lengths? It turns out that the extra cost of a certain scheme may be offset by a special benefit, such as improved convergence rate, better stability with respect to small changes in initial conditions, or strict conservation of quantities that are conserved in the target differential equation. Even if each "turn of the crank" should be more expensive, a scheme may still be competitive if many fewer "turns" are needed for the same result. Or there may be schemes that offer different tradeoffs in the space and time required for a computation (see Bennett [5] and Li and Vitányi [23]), and from among which one would choose on the basis the relative "costing" of storage and processing resources.

In this paper we have introduced *one new type of computational scheme tradeoff*, that between the one-off cost of an efficient but complex infrastructure and the daily drain of energy (or entropy) needed to make a simpler, cheaper infrastructure do the same job. The spacetime regularity of a CA's dynamics explicitly forces one to obey the golden rule, "do unto others as you would have them do unto you". That is, to improve the thermodynamic efficiency of its own computational task, one CA site cannot just borrow some of its right neighbor's state as a scratchpad, lest its left neighbor claim some of *his* state—a vicious infinite regress. But the resources of many CA sites can be pooled together into an LG "supersite". Our results and conjecture deal with how, for any regular CA, a large enough LG supersite can be constructed that achieves *perfect thermodynamic efficiency*. To paraphrase [10], we may term this approach "logic for capitalists".

Having dispensed with a *power supply* for locally lossy CA, if the resulting LG system has to be embedded within an invertible physics one must still complement it with a *heat sink*. Our construction leaves open the issue of how to realize such a sink while retaining the local finiteness of an LG scheme. Can removing and sequestering a finite amount of heat per site be achieved by a finite amount (per site) of extra storage and processing resources by an LG mechanism—and thus by *local* means? This is an area wide open for research.

Finally, there is a rather delicate question that we can barely mention here. We recall that computation is the art of putting together – even though in as large a number of copies as desired – a finite set of logic primitives *given once and for all. Within* any LG this condition is clearly satisfied, as in an LG all processing is done by composition of instances of a *single* unit cell, complex as this may be. But to what extent can one accept as *bona fide computation* an approach that requires designing, for each CA that one wants to rewrite as an LG, new custom primitives, that is, the nodes that make up the LG's very unit cell? Note that these may be arbitrary n-input, n-output Boolean functions.

Among the physical "effects" actually available in nature one may well hope to find one that can be exploited for realizing a simple logic primitive such as the AND gate (or an invertible counterpart thereof, such as the so called "Toffoli gate"

[34,4]). But to what extent will we be able to locate in nature "custom Hamiltonians" capable of realizing (without power assist) arbitrarily complex n-body interactions? Intuitively, information recycling can make computation substantially more efficient (cf. [10]). But, at least within the stylized context of CA and LG, we've shown concrete reasons why recycling schemes may become impractical well before one attains perfect thermodynamic efficiency.

Acknowledgments

References

[1] Serafino Amoroso, Y.N. Patt, Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures, J Comp. Syst. Sci. 6 (1972) 448–464.
[2] Ross Ashby, An Introduction to Cybernetics, Wiley, 1956.
[3] Henry Baker, Linear logic and permutation stacks—the Forth shall be first, ACM SigArch, Comput. Architecture News 22 1 (1994) 34–43.
[4] Adriano Barenco, Charles Bennett, Richard Cleve, David DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John Smolin, Harald Weinfurter, Report on new gate constructions for quantum computation, Phys. Rev. A 52 (1995) 3457–3467.
[5] Charles H Bennett, Time/space trade-offs for reversible computation, SIAM J. Comput. 18 (1989) 766–776.
[6] Cristian Calude, Randomness in cellular automata, in: Sect. 9.5 of Information and Randomness—An Algorithmic Perspective, 2nd ed., Springer, 2002, pp. 367–383.
[7] Silvio Capobianco, Multidimensional cellular automata and generalization of Fekete's lemma, Discrete Math. Theor. Computer Sci. (submitted for publication), arxiv.org/abs/0707.3903.
[8] Jérôme Durand–Lose, Representing reversible cellular automata with reversible block cellular automata, Discrete Math. Theor. Comp Sci. Proc. AA (DM-CCG) (2001) 145–154.
[9] Freeman Dyson, Time without end: Physics and biology in an open universe, Rev. Mod. Phys. 51 (1979) 447–450.
[10] The Economist (staff), Logic for conservatives: The heath-death of a computer, The Economist 11 (1989) 78–79.
[11] M Fekete, On the distribution of roots in certain algebraic equations with integer coefficients, Math. Zeitschr. 17 (1923) 228–249 (in German).
[12] Francesca Fiorenzi, Cellular automata and finitely generated groups, Ph.D. Th. in Mathematics, U. Rome La Sapienza, 2000.
[13] Martin Gardner, The fantastic combinations of John Conway's new solitaire game 'life', Sci. American 223 (1970) 120–123.
[14] J.-Y. Girard, Linear logic, Theoret. Comput. Sci. 50 (1987) 1–101.
[15] Donald Greenspan, Computer-Oriented Mathematical Physics, Pergamon, 1981.
[16] M. Hénon, On the relation between lattice gases and cellular automata, in: R. Monaco (Ed.), Discrete Kinetic Theory, Lattice Gas Dynamics and Foundations of Hydrodynamics, World Scientific, Margolus, 1989, pp. 160–161.
[17] Jarkko Kari, Reversibility of 2D cellular automata is undecidable, Physica D 45 (1990) 379–385.
[18] Jarkko Kari, Representation of reversible cellular automata with block permutations, Math. Syst. Theory 29 (1996) 47–61.
[19] Jarkko Kari, On the circuit depth of structurally reversible cellular automata, Fund. Inform. 38 (1999) 93–107.
[20] Jarkko Kari, Theory of cellular automata: A survey, Theoret. Comput. Sci. 334 (2005) 3–33.
[21] Rolf Landauer, Irreversibility and heat generation in the computing process, IBM J. Res. Dev. 5 (1961) 183–191.
[22] Craig Lent, P. Douglas Tougaw, Wolfgang Porod, Bistable saturation in coupled quantum dots for quantum cellular automata, Appl. Phys. Lett. 62 (1993) 714–716.
[23] Ming Li, Vitányi Paul, An Introduction to Kolmogorov Complexity and its Applications, 2nd ed., Springer, 1997, xx+637.
[24] Douglas Lind, Marcus Brian, An Introduction to Symbolic Dynamics and Coding, Cambridge, 1995, p. xvi+495.
[25] Douglas Lind, Multi-dimensional symbolic dynamics [40, 61–79], Proc. Symp. Appl. Math. 60 (2004).
[26] Norman Margolus, Physics-like models of computation, Physica 10D (1984) 81–95.
[27] Akira Maruoka, Kimura Masayuki, Injectivity and surjectivity of parallel maps for cellular automata, J. Comp. Syst. Sci. 18 (1979) 47–64.
[28] Edward Moore, Machine models of self-reproduction, Proc. Symp. Appl. Math. 14 (1962) 17–33.
[29] John Myhill, The converse of Moore's Garden-of-Eden theorem, Proc. Amer. Math. Soc. 14 (1963) 658–686.
[30] D Richardson, Tessellation with local transformations, J. Comp. Syst. Sci. 6 (1972) 373–388.
[31] John Strickwerda, Finite Difference Schemes and Partial Differential Equations, Chapman & Hall, 1989.
[32] Tommaso Toffoli, Computation and construction universality of reversible cellular automata, J. Comp. Syst. Sci. 15 (1977) 213–231.
[33] Tommaso Toffoli, Cellular automata mechanics, Ph.D. Thesis, The University of Michigan, Comp. Comm. Sci. Dept., 1977.
[34] Tommaso Toffoli, Reversible computing, in: de Bakker, van Leeuwen (Eds.), Automata, Languages and Programming, in: Lecture Notes in Comp. Sci., vol. 85, Springer, 1980, pp. 632–644.
[35] Tommaso Toffoli, Margolus Norman, Cellular Automata Machines—A New Environment for Modeling, MIT Press, 1987.
[36] Tommaso Toffoli, Margolus Norman, Invertible cellular automata: A review, Physica D 45 (1990) 229–253.
[37] Tommaso Toffoli, Capobianco Silvio, Mentrasti Patrizia, How to turn a second-order cellular automaton into a lattice gas: A new inversion scheme, Theoret Comput. Sci. 325 (2004) 329–344.
[38] Tommaso Toffoli, A pedestrian's introduction to spacetime crystallography, IBM J. Res. Dev. 48 1 (Jan 2004) 13–29 (Special issue on Physics of Information).
[39] J.H. van Lint, R.M. Wilson, A Course in Combinatorics, Cambridge, 1992.
[40] Williams Susan (Ed.), Symbolic Dynamics and its Applications, in: Proc. Symp. App. Math., vol. 60, Am. Math. Soc., 2004, p. 168.
[41] Stephen Wolfram, Theory and Applications of Cellular Automata, World Scientific, 1986.
[42] Stephen Wolfram, A New Kind of Science, Wolfram Media, 2002.