Physica D 45 (1990) 229–253
North-Holland

PHYSICS AND CHEMISTRY

INVERTIBLE CELLULAR AUTOMATA: A REVIEW

Tommaso TOFFOLI and Norman H. MARGOLUS
MIT Laboratory for Computer Science, Cambridge, MA 02139, USA

Received 15 January 1990

In the light of recent developments in the theory of invertible cellular automata, we attempt to give a unified presentation of the subject and discuss its relevance to computer science and mathematical physics.

1. Introduction

1.1. Preliminaries

One of the goals of computer science is to provide abstract models of concrete computers, i.e., of whatever computing apparatus can ultimately be built out of the physical world. In this, one seeks *expressiveness* (whatever aspects of the computer are deemed relevant should be captured by the model) and *accuracy* (whatever one can prove about the model should be true about the computer). *Invertible cellular automata* (ICA) are an important development in this direction, of significance comparable to the introduction of Turing machines (and similar paradigms of effective computation) in the late '30s.

Turing machines represent a conscious effort [81] to capture, in axiomatic form, those aspects of physical reality that are most relevant to computation. In this, they are quite unlike Cantor's transfinite sets or similar inventions of the romantic era, and much more like Euclid's principles, which were meant to describe the geometry of our physical world (and one should not forget that, for the Greeks, computation was synonymous with geometrical construction).

Cellular automata are more expressive than Turing machines, insofar as they provide explicit means for modeling *parallel* computation on a spacetime background. However, both classes of models are indifferent to one fundamental aspect of physics, namely *microscopic reversibility*, and thus help create the illusion that, computationally speaking, one lives in a fairy world where the second principle of thermodynamics is not enforced, and "perpetual computation" (in the same sense as "perpetual motion") is possible. By explicitly coming to terms with this aspect of reality, *invertible cellular automata* provide a modeling environment that is more accurate and, in the end, more productive.

Only a few years ago, what was known about ICA could be summarized in a few lines – and was not very exciting either. Today, one can tell a more interesting story, and we shall try to do so in this paper.

Our involvement with ICA represents the convergence of several research trails, including

- Concurrent computation in networks having uniform structure.
- Reversible (or "information preserving") computing processes.
- Fundamental connections between physics and computation.
- Foundations of relativity.
- Quantum computation.
- Fine-grained modeling of physical systems.
- High-performance simulation of cellular automata.
- Data encryption.

1.2. An apology

We have many things to say, and we have a quite varied audience in mind. We want to make

0167-2789/90/$03.50 © 1990 – Elsevier Science Publishers B.V. (North-Holland)

sure that the readers perceive the essential issues before plunging into endless technicalities. For these reasons, we shall follow an informal approach when this seems to enhance clarity, and we shall often present a specific example rather than the most general case.

2. Invertible cellular automata

2.1. Cellular automata

Cellular automata are abstract dynamical systems that play a role in discrete mathematics comparable to that played by partial differential equations in the mathematics of the continuum. In terms of structure as well as applications, they are the computer scientist's counterpart to the physicist's concept of a 'field' governed by 'field equations'. It is not surprising that they have been reinvented innumerable times under different names and within different disciplines. The canonical attribution is to Ulam and von Neumann [82,84] (circa 1950). [#1]

Concise formal definitions are given in section 2.4 (for a more complete formal treatment, see refs. [68,79]). Intuitively, a *cellular automaton* is an indefinitely extended network of trivially small, identical, uniformly interconnected, and synchronously clocked digital computers.

More specifically, we start with an indefinitely extended n-dimensional lattice which represents "space" (typically, n equals 1, 2, or 3 in physical modeling applications). To each site of this lattice, or *cell*, there is associated a state variable, called the *cell state*, ranging over a finite set called the *state alphabet* (typically, the cell state consists of just a few bits' worth of data).

"Time" advances in discrete steps; the dynamics is given by an explicit recipe, called the *local map*, which is used at every time step by each cell to determine its new state from the current state of certain cells in its vicinity.

The local map itself is the composition of two operators, namely, the *neighborhood*, which specifies *which* cells affect the given cell, and the *table*, which specifies *how* those cells affect it. In more detail:

(i) The *neighborhood* lists the relative positions, with respect to the generic cell (in this context also called *center cell*), of a finite number of cells called the cell's *neighbors*. (The neighbors of a cell need not coincide with the cell's "first neighbors" in the array. They may include the cell itself or cells that are several sites away, while cells in between may be skipped. All that is required is that they be finite in number, and be arranged in the same spatial pattern with respect to each cell.)

(ii) The *table* takes the states of a cell's neighbors as arguments, and returns as a result the cell's corresponding new state.

Thus, a cellular automaton's laws are *local* (no action-at-a-distance) and *uniform* (the same rule applies to all sites at all times); in this respect, they reflect two fundamental aspects of physics. Moreover, the system's laws are *finitary*, that is, by means of the local map one can explicitly construct *in an exact way* the forward evolution of an arbitrarily large portion of a cellular automaton through an arbitrary length of time, all by finite means. It is such strong effectiveness built in their definition that makes dynamical systems based on cellular automata appealing to the computer scientist. In continuous dynamical systems, such as those defined by differential equations, the state variables range over an uncountable set, and one has to accept a weaker standard for "effectiveness"; i.e., a specification of the dynamics is accepted as effective if the state of any finite portion of the system at any future time can be computed with an *arbitrarily small error* by finite means. In cellular automata, we demand and obtain zero error.

In this sense, cellular automata present themselves as a finitary alternative to the methods of the calculus in the modeling of spatially extended systems.

2.2. Invertibility

An assignment of states to all cells, i.e., a state for the entire cellular automaton, is called a *configuration*. By applying the local map to every

[#1] At about the same time but quite independently, Zuse [91] proposed structures, intended as digital models of mechanics, that are essentially cellular automata.

site of the array, from any configuration q one obtains a new configuration q', called its *successor*. Thus, the local map defines a transformation $q \overset{\tau}{\mapsto} q'$, called the *global map*, on the set of configurations.

A cellular automaton is *invertible* if its global map is invertible, i.e., if every configuration – which, by definition, has exactly one successor – also has exactly one predecessor.

In the context of dynamical systems, invertibility coincides with what the physicists call 'microscopic reversibility'. This should not be confused with 'invariance under time reversal', which is a stronger property. #2

Initially, cellular automata were used chiefly as "toy models" for phenomenology associated with dissipative (i.e., macroscopically irreversible) processes; typical topics were biological organization [40,23], self-reproduction [84], chemical reactions, and visual pattern processing [59]. Since the interest was more in exploring the *consequences* of irreversible behavior rather than its *origins*, it was not only harmless but actually expedient (given the severely limited computing resources) to use models where irreversibility happened to be built-in. #3 Thus, it is not surprising that no need was felt for ICA.

It should be noted that the issue of invertibility wasn't even present in the minds of most cellular automata investigators. To the few to whom it was, it wasn't at all clear whether ICA could actually lend themselves to the modeling of microscopically reversible physics.

The perceived difficulties were of two kinds. On one hand, there were no practical procedures known for constructing nontrivial ICA; on the other, it was suspected and argued that ICA did not have adequate computing capabilities.

Both of these difficulties have now been amply removed. Indeed, ICA have become an important tool of computational physics in applications where the explicit modeling of reversible phenomena is concerned. Moreover, they are playing an increasingly important role as conceptual tools of theoretical physics.

2.3. Historical notes

As already mentioned, cellular automata were initially treated as some sort of conceptual erector set – a plaything for interdisciplinary biologists and computer scientists – and drew little attention from professional mathematicians. This may explain why the issue of invertibility – which in mathematical systems theory is one of obvious priority – was slow to be explicitly recognized by the cellular automata community.

In 1962, Moore [51] asked whether there could exist "Garden of Eden" configurations – i.e., configurations that do not have a predecessor – and proved that, under certain conditions, if a configuration has more than one predecessor then there must be a configuration that has none [51]. The converse was proved by Myhill [53] in 1963.

Moore's and Myhill's results originated a lengthy "Garden of Eden" debate (see references in ref. [61]), which brought to light a number of subtle issues somehow related to invertibility. But invertibility was explicitly addressed only in 1972, in seminal papers by Richardson [60] and Amoroso and Patt [2]. #4

After that, theoretical work on invertibility in cellular automata proliferated (see refs. [3,61,54] and [46–48,90,35]). In spite of that work, however, for many years the most interesting ICA actually exhibited remained an extremely simple-minded one (the longest orbit is of period two!), discovered by Patt through brute-force enumeration [56]. ICA continued to "appear to be quite rare" [2].

Not only rare, but also simple-minded! On the

#2 Let $\tau: Q \to Q$ be an invertible dynamical system. A bijective mapping $\phi: Q \to Q$ obeying appropriate regularity properties (e.g., continuity, translation invariance, etc., depending on the context) is called a *time-reversal* operator if $\tau^{-t} = \phi^{-1}\tau^t\phi$; the system is *invariant under time reversal* if it admits of such an operator. Thus, a time-reversal invariant system is not only invertible but also *isomorphic to its inverse*. Hamiltonian mechanics has the well-known time-reversal operator $\phi: \langle q,p\rangle \mapsto \langle q,-p\rangle$.

#3 This interest is not abating; see, for instance, refs. [55,7,8,10].

#4 Unbeknownst to those authors, systems that are in essence one-dimensional cellular automata had already been studied in an abstract mathematical context by Hedlund and associates as early as 1963 [30,31]; both Richardson's results on invertibility (section 4.3) and Patt's search for ICA (section 5.3) had been anticipated by Hedlund's school.

basis of various kinds of circumstantial evidence (cf., e.g., ref. [63]), Burks conjectured that an ICA cannot be computation-universal [11] (note that that was at a time when computation universality was being turned up under almost every stone), and soon Aladyev [1] appeared to have proved Burk's conjecture.

Finally, except for the one-dimensional case [2], no one even knew of a systematic procedure for telling whether or not a cellular automaton was invertible.

In summary, for a long time ICA seemed to lack any appeal or promise.

Following fundamental results by Bennett on invertible Turing machines [6], in 1976 one of us (Toffoli) proved the existence of ICA that are computation- and construction-universal [67], and noted the relevance of this, in principle, to the modeling of physics [68].

In the same year, and unnoticed by the (then meager) cellular automata establishment, Pomeau [26] discussed, as a model for hydrodynamics, a "lattice gas" that is in fact an ICA and a useful stylization of certain microscopically reversible physical interactions.

Independently, Fredkin had been studying invertible recurrences as models of dynamical behavior; had arrived at techniques for synthesizing arbitrary sequential behavior out of invertible Boolean primitives [19]; and had studied a class of ICA (see section 5.4) that displayed some analogy with Lagrangian mechanics.

Rapid, synergistic developments finally started taking place in the early '80s.

In 1981, a conference on "Physics and Computation" [20] explicitly addressed the theme of *fundamental* connections between physics and computer science (rather than more incidental ones, such as computer programs for the numerical integration of differential equations). Ideas such as "virtually nondissipative computation" [19] and "quantum computation" [5,18] started gaining legitimacy.

The links between a number of physicists and computer scientists interested in these themes were tightened by a follow-up workshop on Moskito Island (1982), where an early prototype of cellular automata machine was demonstrated by us.

Wolfram's 1983–1986 sortie into the cellular automata arena [85–89], stimulated by that workshop, was in turn a determining factor in introducing a generation of mathematical physicists to the cellular-automaton paradigm.

Inspired by Fredkin's "billiard-ball" model of computation [19], Margolus arrived in 1983 at a very simple computation-universal ICA[41] that is suggestive of how a computer could in principle be built out of microscopic mechanics. At about the same time, Vichniac [83] and Creutz [13] pioneered the use of cellular automata for the microcanonical modeling of Ising spin systems.

The introduction of dedicated cellular automata machines [71] encouraged much new experimental work on ICA, and stimulated further theoretical developments.

For instance, according to Pomeau, seeing (at a second Moskito Island workshop, in 1984) his lattice-gas model running on one of these machines made him realize that what had been conceived primarily as a *conceptual* model could indeed be turned, by using suitable hardware, into a *computationally accessible* model. This stimulated his interest in finding lattice-gas rules that would provide better models of fluids. In the past few years, lattice-gas hydrodynamics has grown into a substantial scientific business (see section 5.6).

In turn, the growing interest in fine-grained models of physics and in their potential applications to important practical problems created the need for computers capable of handling large models of this kind much more efficiently than conventional scientific computers [45]. A second generation of cellular automata machines, whose development is almost complete [44,78], reflects in its architecture the objective to efficiently support the simulation of ICA– which are likely to constitute a major portion of its fare.

2.4. Terminology

The present section complements with precise definitions and notation the informal terminology introduced in sections 2.1–2.2. Refer to refs. [31,72] for more abstract, but equivalent, definitions given in terms of continuity in the Cantor-set topology.

Space. Let $S = Z^n$ denote the Abelian group of translations of an n-dimensional lattice I onto itself. It will be convenient to call the elements of S *displacements*. The sum and the difference of two displacements are again displacements. The application of a displacement $s \in S$ to a site $i \in I$ yields a new site denoted by $i + s$. The *difference* $i' - i$ between two sites is the displacement s such that $i' = i + s$.

Interconnection. A *neighborhood* is a finite set of displacements (i.e., a subset of S). Typically, a neighborhood X is applied as an operator to a site i, yielding a set of sites. That is, the X-*neighborhood* (or simply the *neighborhood*, when X is understood) of i is the set $i+X = \{i+x | x \in X\}$; the elements of $i + X$ are the *neighbors* of i, and are naturally indexed by the elements of X.

The *radius* of X is the length of its longest element. #5

State. Given a lattice I and a nonempty state alphabet A, the set Q of *configurations* (of A over I) is the Cartesian product of copies of A indexed by the set I, i.e., $Q = A^I$. The ith component, in this product, of a configuration q is called the state of site i in configuration q, and is denoted as usual by q_i. Note that $q_i \in A$ can be thought of as the result of applying to configuration $q \in A^I$ the projection operator $[i]$ associated with the ith coordinate of the Cartesian product, i.e., $q_i = [i]q$.

More generally, a *neighborhood projection* operator $[i+X]$ will extract from a configuration q the collective state of the neighbors of i, denoted by $[i + X]q$ or q_{i+X}. Note that $q_{i+X} \in A^X$.

Dynamics. A *local map* is a pair $\lambda = \langle X, f\rangle$, where X is a neighborhood and f a *table*, i.e., a function of the form $f\colon A^X \to A$. The table f can be applied to any site i of a given configuration q through the agency of the neighborhood projection operator, which will extract from q and supply to f the correct set of arguments. Let q'_i be the result of this application, i.e.,

$$q'_i = f q_{i+X} \;\; (= f[i + X]q). \tag{1}$$

The symbol q'_i can be interpreted as the state at site i of a new configuration q'. The relation $q' = \tau q$ defines a new function $\tau\colon Q \to Q$, called the *global map* induced by the local map λ.

Note that the local map can be thought of as a function $\lambda\colon \langle Q, I\rangle \to A$ defined (cf. (1)) by

$$\lambda(q, i) = \langle X, f\rangle(q, i) = f q_{i+X}. \tag{2}$$

The sequence of configurations obtained from an initial configuration q^0 by iterating the map τ will be denoted by

$$q^0, q^1, q^2, \ldots \tag{3}$$

where the superscript represents the sequence index rather than an exponent.

A table f, formally given as a function of k arguments (k is the size of the neighborhood X), may happen to depend vacuously on some of these arguments. In that case, one can maximally reduce neighborhood and table in an obvious way, yielding the *effective* neighborhood and the corresponding table – which together make up the *reduced* local map. Unless otherwise noted, we shall tacitly assume that local maps are given in reduced form.

3. Universality

Little needs to be added here on the universality theme.

In ref. [67], the computation universality of ICA was proved by showing that every computation-universal cellular automaton can be embedded in an invertible one having one more dimension. This left open the question of whether *one*-dimensional ICA could be computation-universal. A positive answer was recently given by Morita and Harao [50].

The constructions of refs. [67,50] are more concerned with existence than with efficiency. More direct constructions can be more instructive as well as more efficient. Indeed, if one wants to build a general-purpose computing structure within a cellular automaton, the most practical approach is to start with a local map that directly supports logic gates and wires, and then build the appropriate logic circuits out of these primitives [4,77]. As explained in refs. [70,19], in an *invertible* cellular automaton the gates will have to be invertible; because of this constraint, a complete, self-contained

#5 For our purposes, the Euclidean metric will do, even though it is an overkill.

logic design will have to explicitly provide, besides circuitry for the desired logic functions, additional circuitry for functions (analogous to energy supply and heat removal in ordinary computers) concerned with entropy balance. This issue, which had been bypassed in ref. [67], is directly addressed by ICA such as the BBM model devised by Margolus [41].

4. Decidability

One of the first questions to come to mind is, of course, "How does one tell if a cellular automaton is invertible?" We shall now present the essential aspects of this question; further details will be given in the following sections.

4.1. A parable

> FOR SALE: Local map λ of invertible cellular automaton SPRIZE. Interesting behavior, lots of fun. $29.95. Call John at ×3194.

This ad catches my attention. I already have a bunch of cellular automata at home, that I can run on my personal computer, but this one claims to be invertible: its global map τ has an inverse τ^{-1}, and by running τ^{-1} I can watch the automaton go backwards in time! I send my check. Four days later I receive a diskette containing a data file `SPRIZE.LOC` (which, I presume, tabulates the local map λ of SPRIZE). I bring up my cellular-automaton simulation program, load `SPRIZE.LOC`, initialize the screen with a blob of random junk on a clear background, and hit the RUN key. The blob starts churning – perhaps it's spreading a bit – yes, it's spreading, but very slowly – rather boring, I'd say. Wait – look at that long caterpillar crawling up the screen! How the heck did it get started?

Can I go backwards in time and see exactly how the caterpillar emerged out of the random blob? I look in the diskette directory, and I find a second file labeled `-SPRIZE.LOC`. That must be it – the local map $\bar{\lambda}$ of SPRIZE's inverse! I load it. There goes the caterpillar crawling backwards, curling up into some sort of cocoon at the edge of the blob, and finally dissolving into randomness!

I make a few more experiments. Indeed, the cellular automaton SPRIZE defined by the local map λ is invertible and $\bar{\lambda}$ is the local map of its inverse.

That's a happy conclusion – but the story could have ended differently.

Suppose I didn't find a file `-SPRIZE.LOC`. What good is knowing that SPRIZE is invertible if I don't have an effective way to run its inverse? Worse yet, in this situation how can I be sure that John's ad was truthful – that SPRIZE *is* invertible? Can I prove it? Or perhaps disprove it and claim a refund?

Assuming that SPRIZE is after all invertible, can I find $\bar{\lambda}$ by myself, starting from λ? How long will that take? By definition, $\bar{\lambda}$ is a finite object, and I can sequentially generate all possible candidates. But how would I recognize the right one? And is the very existence of $\bar{\lambda}$ guaranteed? In other words, if the global map τ has a local description, does it follow that also its inverse τ^{-1} has a local description – that the inverse of a cellular automaton is again a cellular automaton?

4.2. The finitary connection

To summarize, for any given cellular automaton we would like to know whether or not it is invertible; if it is, we would like to have a finite recipe also for its backward evolution, i.e., the *inverse* local map – as contrasted to the *direct* local map of the forward process.

For the sake of comparison, let's look at a dynamical system defined by a differential equation. Consider, for example, the evolution of the temperature distribution $q(x,t)$ along a metal bar, according to the "heat" equation

$$\frac{\partial q}{\partial t} = \frac{\partial^2 q}{\partial x^2}. \tag{4}$$

This equation can be thought of as a local recipe for an "infinitesimal" forward step,

$$q(t+\mathrm{d}t)|_x = q(t)|_x + \left.\frac{\partial^2 q}{\partial x^2}\right|_x \mathrm{d}t \tag{5}$$

and by just turning +'s into −'s one immediately obtains a local recipe for an infinitesimal backward step. Thus, whenever the forward evolution is defined, so is the backward one, and a local inverse recipe is known as soon as a direct one is. In conclusion, with differential equations there is an immediate connection between direct and inverse local recipes. (However, one should keep in mind that, as noted in section 2, these local recipes are of a less effective kind than those of cellular automata.)

On the other hand, with ICA, the only connection we have in principle between direct and inverse local maps is through a *non-finitary* route, as is illustrated by the following diagram:

$$\begin{array}{ccc} \textit{finitary} & & \textit{non-finitary} \\ \textit{constructs} & & \textit{constructs} \\ \text{direct local map} & \longrightarrow & \text{direct global map} \\ & & \downarrow \\ \text{inverse local map} & \longleftarrow & \text{inverse global map} \end{array} \qquad (6)$$

Under what conditions can we establish a *finitary* route from a direct local map to an inverse one?

4.3. The fundamental lemmas

Whatever the technical difficulties in charting such a route, it is important to know that the endpoint exists and is recognizable, as shown by the following two lemmas.

Lemma 4.1 (Richardson [60]) *If a cellular automaton is invertible, then its inverse is a cellular automaton.*

This is a fundamental result. It tells us that if the global process described by a local map is invertible, then also the inverse global process has a local map, i.e., it can be described in local terms. On the other hand, the proof (which is based on topological arguments of a general nature) doesn't give any explicit method for constructing the inverse local map.

Lemma 4.2 *There is an effective procedure for deciding, for any two local maps λ and λ' defined on the same set of configurations, whether the corresponding global maps τ and τ' are the inverses of one another.*

Proof. The composition $\lambda'' = \lambda'\lambda$ is a new map operating on the neighborhood consisting of the neighbors (according to λ) of the neighbors (according to λ') of the generic cell. If for any value a of this cell the map λ'' returns the value a, independently of the values of the other neighbors, then $\tau'\tau$ is obviously the identity function. If a value different from a is returned from some choice of values for the other neighbors, then $\tau'\tau$ obviously differs from the identity. □

The construction in the above proof provides a way to verify whether an alleged inverse local map λ' of a direct local map λ is indeed such an inverse. This test is our touchstone for certifying that a cellular automaton is invertible. Whether the candidate λ' was generated by a rigorous construction, suggested by heuristic methods, dictated by an oracle, encountered on a search, or even arrived at by faulty arguments, if it passes the test it can be accepted without further question.

Conceivably, it might be possible to prove the invertibility of a cellular automaton without exhibiting a local map for its inverse. However, we don't know of any cases where this has been done. For all of the ICA known today, the inverse local map comes "bundled", as it were, with the direct one.

4.4. The fundamental theorems

From lemma 4.2 one immediately obtains

Theorem 4.3 *The class of invertible cellular automata is recursively enumerable.*

Proof. Sequentially generate all local maps, say, in order of increasing complexity (measured in terms of state-alphabet size and neighborhood radius). For each item λ in this sequence, start a new enumeration of all local maps, and for each item λ' of this enumeration test whether λ' is the inverse local map of λ. Each match yields an ICA, and every ICA will eventually turn up in the course of this double enumeration. □

In other words, invertibility in cellular automata is at least *semi*decidable: if a specific cellular automaton is invertible, the above procedure will positively let us know; however, if it is not invertible, at no moment in time will we be positively told of that.

How about *full* decidability? One partial result was already mentioned in section 2.3, namely,

Theorem 4.4 (Amoroso and Patt [2]) *There is an effective procedure for deciding whether or not an arbitrary one-dimensional cellular automaton, given in terms of a local map, is invertible.*

Amoroso and Patt thought that the techniques employed by them 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".

Since then, for almost twenty years a quest for these "generalizations" to more than one dimension went on with little success. (Invertibility and related properties for the one-dimensional case were revisited in [54,87,14,29].) Many equivalent characterizations of ICA were given [90,47,48,35], but none that offered a finitary handle on invertibility.

Finally, quite recently, Kari proved that

Theorem 4.5 (Kari [38,39]) *There is no effective procedure for deciding whether or not an arbitrary two-dimensional cellular automaton, given in terms of a local map, is invertible.*

His proof is by reduction to the undecidability of the tiling problem, and depends on the availability of a set of tiles having certain properties. He exhibits such a set, but the construction runs over fifteen pages; given the importance of the result, we hope that a shorter proof will be found soon.

Thus, the invertibility of a cellular automaton is, in general, undecidable. This has important implications, some of which we intend to discuss (section 8). But, lest the readers feel that they are groping completely in the dark, let us first balance the negative result of theorem 4.5 with a body of positive results concerning ICA.

5. Ways to make invertible cellular automata

Suppose one wanted to get a general feeling for what kinds of behavior are possible with ICA. One could start with a good assortment of these automata, and study a number of cases in detail. The problem is, how does one put together such an assortment?

It turns out that of all cellular automata the invertible ones constitute a vanishingly small subclass [62]. Moreover, as we have seen in section 4, there is no effective procedure for determining whether or not an arbitrary cellular automaton (as specified by a local map) is invertible. Finally, even if one were willing to fall back on a brute-force search (theorem 4.3), a long search time would generate only a few items, and even those would be for the most part quite uninteresting.

In our experience, rather than spend an inordinate amount of resources on a blind search for these objects that are rare, hard-to-recognize, and more often than not quite plain, it is more rewarding to attempt the direct synthesis of special cases having certain desirable features. In this section, we shall discuss a number of synthesis techniques which yield a rich variety of ICA.

5.1. Trivial cases

Let us consider the following two cases:

(a) *Each cell is allowed to look only at itself as a neighbor.* Then the cellular automaton reduces to a collection of finite, isolated systems – one per cell.

(b) *Each cell is instructed to copy, say, its* left *neighbor.* Then the whole configuration will shift one cell to the *right* at each time step. This 'uniform shift' behavior can be factored out of the dynamics by a simple coordinate transformation of the form $x \mapsto x - t$. After this transformation, the global map reduces to the identity, and again the cellular automaton collapses into a collection of finite, noninteracting systems.

A cellular automaton whose neighborhood consists of at most one cell, as in (a) or (b) above, is called *trivial* [2].

Clearly, a trivial cellular automaton is invertible if and only if its effective neighborhood X consists of *exactly one* element, and its table, of the format $A \xrightarrow{f} A$, is invertible, i.e., is a *permutation* of the state alphabet A.

5.2. The incredible shrinking neighborhood

It would be nice if the invertibility of a cellular automaton could always be reduced, as in the previous section, to the invertibility of its table. Unfortunately, when the neighborhood X consists of more than one element, the table f as such cannot be invertible, since the two sets that appear in $A^X \xrightarrow{f} A$ have different cardinalities.

Let us concentrate on this point. From the viewpoint of the global map, $q^{\text{old}} \overset{\tau}{\mapsto} q^{\text{new}}$, the configuration q^{old} contains all the information needed to construct q^{new} (via τ); if the automaton is invertible, also q^{new} contains all the information needed to construct q^{old} (via τ^{-1}). This symmetry between the two directions of time is not preserved, in general, when the same dynamics is expressed by a local map $q^{\text{old}}_{i+X} \overset{f}{\mapsto} q^{\text{new}}_i$; in fact, while the new state of a cell is completely determined by the old state of its neighbors, the latter is not completely determined by the former – *even if the cellular automaton is invertible.* [#6]

Yet, the only way we know to construct ICA is to somehow manage to overcome the above difficulty of format, and in the end express the local map in terms of permutations of the state alphabet. Different ways of doing this are presented in the next three sections; here we'll try to capture the flavor of this approach.

In a trivial ICA (cf. previous section), a new configuration is obtained from an old one by applying a given permutation of the state alphabet, π, to each cell, i.e., $q^{\text{old}}_i \overset{\pi}{\mapsto} q^{\text{new}}_i$; of course, π itself is independent of i. Consider now the set Π of *all* permutations of the state alphabet, and make a dynamical system where each site i uses a permutation $\pi^t_i \in \Pi$ that may be *different* from site to site and from moment to moment, i.e.,

$$q^{t+1}_i = \pi^t_i q^t_i. \tag{7}$$

This system is not a cellular automaton, because its dynamics is space- and time-dependent, but is still invertible. In fact, since the π^t_i are assigned once and for all for each i and t, the inverse system is explicitly given by

$$q^t_i = (\pi^t_i)^{-1} q^{t+1}_i. \tag{8}$$

Now, the trick to restore space- and time-invariance is to make the choice of π^t_i depend on i and t not directly, but only indirectly, as a function of the "landscape" that can be seen from site i at time t (i.e., the state of the neighborhood not including the center cell itself). Now we are back to a cellular automaton, but, unless we are careful, we may lose invertibility, since the landscape itself will in general change under the action of the local map. All that is left to do is make sure of the following: If the "old" landscape of site i told us (by means of a definite recipe ρ) to use permutation π^t_i at site i and time t while going forward in time, then the "new" landscape of site i must be able to point (by means of a matching recipe $\bar{\rho}$) at the same permutation π^t_i (which we will then invert) when going backwards in time.

In section 5.3, this is done by making sure that the relevant landscape does not change at all, so that π^t_i does not in fact depend on t. In section 5.4, only half of the state variables are allowed to change at each step, while the other half, which does not change, provides a landscape that is recognizable from either direction of time-travel; the two halves exchange roles after each step. Finally, in section 5.5, center cell and landscape are fused into an indivisible block of cells, and the dynamics is effectively given by a permutation that acts on the entire block rather than on a single cell.

5.3. Conserved-landscape permutations

In 1971, Patt [56] conducted a search for nontrivial ICA, restricting himself to one dimension (where a decision procedure is known), two states per cell, and contiguous neighbors. He found none for neighborhoods of size 2 and 3. His search stopped at size 4, where he found exactly eight cases (out of 65,536). All the eight cases are variants (obtained by reflection or complementation) of a single cellular automaton, described by the following local map (where the center cell is underscored):

$$\begin{array}{llll} 0\underline{0}00 \mapsto \underline{0} & 0\underline{1}00 \mapsto \underline{1} & 1\underline{0}00 \mapsto \underline{0} & 1\underline{1}00 \mapsto \underline{1} \\ 0\underline{0}01 \mapsto \underline{0} & 0\underline{1}01 \mapsto \underline{1} & 1\underline{0}01 \mapsto \underline{0} & 1\underline{1}01 \mapsto \underline{1} \\ 0\underline{0}10 \mapsto \underline{1}* & 0\underline{1}10 \mapsto \underline{0}* & 1\underline{0}10 \mapsto \underline{0} & 1\underline{1}10 \mapsto \underline{1} \\ 0\underline{0}11 \mapsto \underline{0} & 0\underline{1}11 \mapsto \underline{1} & 1\underline{0}11 \mapsto \underline{0} & 1\underline{1}11 \mapsto \underline{1}. \end{array} \tag{9}$$

[#6] Intuitively, the neighbors of site i will also affect sites other than i; in turn, when time is made to flow backward, they may be affected by sites other than i.

What makes this cellular automaton invertible? Note that the local map specifies 'no change' except for those two entries (starred in the table) where the center cell is surrounded by the landscape 0•10 – in which case the cell itself invariably "flips", i.e., complements its state. What property of this landscape is crucial to the automaton's invertibility? #7

It turns out that, with local map (9), changing the state of the center cell in the landscape 0•10 cannot lead to the creation or the destruction of other occurrences of that landscape – the landscape is *conserved.* This is easier to verify if the local map, of the form

$$q_i^{\text{new}} = \pi_i q_i^{\text{old}}, \tag{10}$$

is explicitly tabulated as follows ('–' denotes a "don't care" argument)

landscape(i)	π_i
– • – 1	'no change'
0 • 1 0	'flip'
– • 0 –	'no change'
1 • – –	'no change'

In these circumstances, if one attempted to "undo" one step of the dynamics, one would know exactly *which* cells have just been changed (because wherever a change was permitted the corresponding landscape was conserved), and *how* to undo the changes (because the effect of a 'flip' can be reversed by flipping again). In fact, the inverse local map for this dynamics is identical to the direct one.

Note that a conserved landscape prevents most of the cells of a configuration from ever changing state; at a higher hierarchical level, those cells can be regarded as structural parameters of the machinery rather than state variables. The remaining cells, which constitute the effective state variables, are decoupled from each other, leading to a situation where invertibility is determined in a trivial way, much as in the previous section.

#7 Notice that when the flip is conditioned by a different landscape, the resulting cellular automaton is not, in general, invertible. For instance, with the rule where the center cell flips in the landscape 1•00, the configuration ...00000100000... has no predecessors.

Invertibility in conserved-landscape cellular au tomata is not limited to trivial cases. For exan ple, by making use of several conserved landscape that partially overlap one another, one can sele tively retain some coupling between cells; in pa ticular, one can construct ICA that simulate an second-order ICA (cf. section 5.4).

5.4. Second-order cellular automata

We shall give a simple method for obtainin starting from an arbitrary cellular automaton, new one that is invertible and has a neighborhoo at least as large as that of the original – and th is nontrivial if the original was nontrivial.

To paraphrase Zeno, if we cut a single frame ou of the movie of a flying bullet, we have no wa of knowing what the bullet is doing. However, we are given two consecutive frames, then we ca figure out the bullet's trajectory. That is, fro these two frames, interpreted as the bullet's "past and "present" positions, we can construct a thi frame giving the bullet's "future" position; th procedure can be iterated.

The laws of Newtonian mechanics happen to b such that, if for some reason the two frames g exchanged, we would end up figuring the bullet trajectory *in reverse.* The present approach to i vertibility in cellular automata, suggested by E Fredkin of MIT, is based on the above mechanic metaphor.

Let us start with a dynamical system in whi the sequence of configurations that make up a tr jectory is given by a relation of the form

$$q^{t+1} = \tau q^t. \quad (1$$

For the moment, we can think of the q^t as real va ables. In general, (11) gives rise to a noninvertib dynamics (i.e., there may be no way, or no uniq way, to extend the sequence backwards). For t dynamics to be invertible, τ itself must be inve ible.

Now, let us consider a new system, defined the relation

$$q^{t+1} = \tau q^t - q^{t-1}. \quad (1$$

This is an example of *second-order* system, whe the "next" configuration is a function of both t

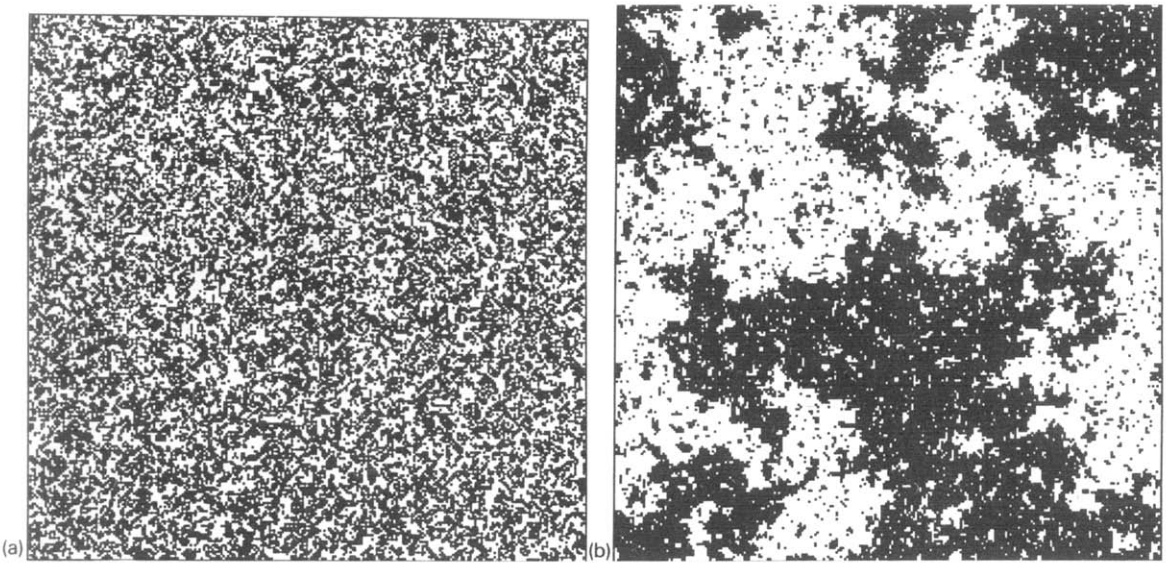

Fig. 1. Equilibrium configurations of Q2R above the critical energy (left) and at the critical energy (right).

"current" and the "previous" one – and thus it takes a *pair* of consecutive configurations to completely determine the forward trajectory. In general, second-order relations give rise to noninvertible dynamics. However, a relation of the specific form (12) guarantees the invertibility of the dynamics for an *arbitrary* τ. In fact, by solving (12) with respect to q^{t-1}, one obtains the relation

$$q^{t-1} = \tau q^t - q^{t+1}; \tag{13}$$

that is, a pair of consecutive configurations suffice to determine in a unique way also the *backward* trajectory.

The above considerations can immediately be applied to cellular automata (see ref. [77, ch. 6] for an intuitive presentation). In equation (11), let the q^t be configurations of a cellular automaton, and τ a global map. The local map will be of the form

$$q_i^{t+1} = f q_{i+X}^t. \tag{14}$$

We shall identify the r elements of the state alphabet A with the integers $0, 1, \ldots, r-1$. Then (12), with "$-$" denoting the *difference* [#8] mod r between two configurations, defines a particular *second-order* cellular automaton whose local map is

$$q_i^{t+1} = f q_{i+X}^t - q_i^{t-1}. \tag{15}$$

Note that the above equation can be rewritten as

$$q_i^{t+1} = \pi_{q_{i+X}} q_i^{t-1}, \tag{16}$$

where the permutation π that turns q_i^{t-1} into q_i^{t+1} changes, from site to site, as a function of the "landscape" q_{i+X}.

With this method, from any ordinary cellular automaton with state alphabet A and global map τ one immediately obtains from (15) a second-order cellular automaton that is invertible. The latter, in turn, can always be written as an *ordinary* cellular automaton, with state alphabet $A \times A$, configurations of the form $\langle q^{t-1}, q^t \rangle$ and global map of the form

$$\langle q^{t-1}, q^t \rangle \mapsto \langle q^t, \tau q^t - q^{t-1} \rangle. \tag{17}$$

Thus, in spite of the great "rarity" of ICA, the ones we can construct are at least "as many" as the noninvertible ones!

Second-order recurrences in which the individual state variables are real numbers (rather than bits) are, of course, routinely used in constructing

[#8] Of course, "$-$" as an operator on configurations is induced from "$-$" as an operator on single cells, as used for instance in (15) below, by applying it in parallel to all sites.

finite-difference schemes for Lagrangian systems, and recurrences of the form (17), in particular, for obtaining invertible dynamics (cf. ref. [12]). Note, however, that – no matter whether the state variables are real numbers or symbols from a finite state alphabet – a permutation operator of the form $\pi_{q_{i+X}}$, as in (16), is much more general than a difference operator of the form '$fq^t_{i+X}-$', as in (15).

We shall briefly present two examples of second-order ICA that combine richness of behavior with economy of means. #9

The **Q2R** rule, introduced by Gérard Vichniac [83] (for more detail, see ref. [77, ch. 17]), is the simplest microcanonical model of a two-dimensional *Ising spin system*. The two elements of the state alphabet $A = \{\uparrow, \downarrow\}$ can be thought of as the two orientations of a spin-$\frac{1}{2}$ particle tied to each lattice site. The neighborhood consists of the four "first neighbors" of a cell (in the four directions of the compass). The operator π in (16) specifies 'flip' if two of the four neighbors are spin-up and two spin-down, and 'no change' in all other cases. If one assigns one unit of potential energy to each occurrence of an antiparallel bond ($\uparrow\downarrow$) between a cell and one of its neighbors, the above rule is equivalent to flipping a spin whenever this operation is energetically indifferent. Thus, the state of the entire spin array moves, in phase space, along a surface of constant energy. #10 Innumerable variants and generalizations of this basic model can, of course, be devised (cf. ref. [77]). But even in this bare form the model is adequate for illustrating the richness and the theoretical challenges of critical phenomena theory (symmetry breaking, phase transitions, long-range correlations, etc.). Fig. 1 illustrates the onset of phase separation as one crosses the critical temperature.

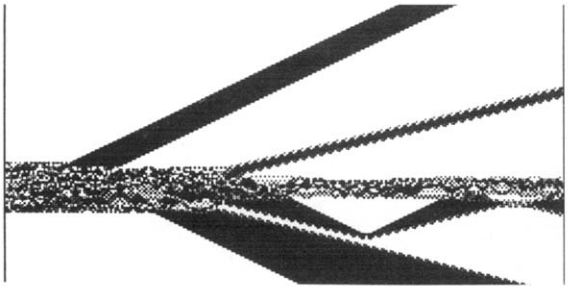

Fig. 2. A spacetime history from the **SCARVES** rule. Time progresses righwards.

The **SCARVES** rule [77, p. 97], introduced and extensively studied by Charles Bennett, is a one-dimensional system; it supports a variety of "elementary particles" that travel at different speeds and undergo various types of interaction, as illustrated in fig. 2. The four neighbors are now the two "first neighbors" and the two "second neighbors" in the one-dimensional array; except for that, the operator π can be described by the same words as that for **Q2R**.

Other simple examples of second-order cellular automata relevant to statistical mechanics are discussed in ref. [77] and in refs. [64,66]; the analogy with Lagrangian and Hamiltonian mechanics is explored further in refs. [43,37].

5.5. *Partitioning cellular automata*

The advantage of trivial cellular automata is that the domain and the range of the table are the same set, so that invertibility of the table implies invertibility of the dynamics; the disadvantage, of course, is that each cell is its only neighbor, so that there is no communication between cells.

Following refs. [41,43], let us try to keep the advantage and remove the disadvantage. At time 0, let us cut up space into finite regions, obtaining a partition p_0 of the set of sites (in fig. 3, the thick lines delimit regions consisting of four-cell squares). Let us introduce a new kind of local map that takes as input the contents of a region and produces as output the new state of the *whole* region (rather than of a single cell). Such a map f_0 allows information to be exchanged between cells

#9 Another example is the earlier cellular-automaton realization (2 dimensions, 3 states, 9 neighbors) of the "billiard ball" model of computation [19], discussed by Margolus in ref. [41, appendix A]. The encouraging results obtained through this construction eventually lead to to a more compact realization of the billiard-ball model via the partitioning technique (section 5.5).

#10 **Q2R** actually consists of two intermeshed but independent sublattices, one evolving in the "white" squares an the other in the "black" squares of a spacetime checkerboard, and each separately conserving energy. In ref. [77], we discuss a number of ways to overcome this redundancy by using cellular automata of slightly different formats.

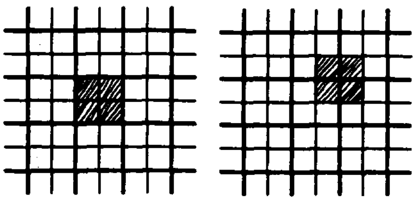

Fig. 3. Even (thick lines) and odd (thin lines) partitions of a two-dimensional array into four-cell blocks. One block in each partition is shown shaded.

of a region, but not across region boundaries. If f_0 is invertible (note that the format of f_0 is "four cells to four cells"), the corresponding global map τ_0 is also invertible. If we kept iterating τ_0, information would remain locked up within each region. Instead, at the next step let us use a *different* partition, P_1 (thin lines in fig. 3), and a local map f_1 of the same kind as f_0, but acting on the regions of P_1. Again, if f_1 is invertible so is the corresponding global map τ_1. The regions of P_1 straddle the boundaries between regions of P_0, and so information that at step 0 had been dammed up within a region of P_0 may at step 1 spill over adjacent regions. We shall keep alternating between τ_0 and τ_1, respectively at even and odd time steps.

We have thus achieved our two main goals, namely, we have constructed a structure, called a *partitioning* cellular automaton (generalizations to larger regions and longer time-step cycles are obvious), in which (a) global invertibility derives in a straightforward way from local invertibility, and (b) information can be transmitted over any distance (i.e., the system is an interacting whole rather than a collection of isolated subsystems). #11

The partitioning technique is particularly useful for constructing systems consisting of moving particles. To "move" a particle in a cellular automaton one must actually *erase* the particle from its current site i and *create* a new copy of it on an adjacent site j. These two operations must be carefully matched, lest particles vanish or multiply; unfortunately, in ordinary cellular automata the information available to the local map when acting on site i (i.e., the state of the neighborhood of i) is different from that on site j, and that makes it difficult for two distinct applications of the local map to carry out the two halves of the *same* decision ('move' or 'not move'). Special handshakes must be devised. #12 With partitioning cellular automata, the two halves of a 'move' operation can be made to fall within the scope of the same neighborhood, and thus coordination is trivial. This feature has an immediate application in lattice gases (section 5.6) and similar interacting particle systems.

In the same way as one insures the conservation of the number of particles, one can insure the conservation of other quantities of physical interest (variables that represent momentum, charge, etc.).

5.6. Lattice gases

Intuitively, a *lattice gas* is a system of particles that move in discrete directions at discrete speeds, and undergo discrete interactions. It will be clear in a moment that a lattice gas is but a special format of cellular automaton; however, it will be useful to start with a example in which the more picturesque terminology of continuous motion is retained.

In the HPP lattice gas [26], identical particles move at unit speed on a two-dimensional orthogonal lattice, in one of the four possible directions. Isolated particles move in straight lines. When two particles coming from opposite directions meet, the pair is "annihilated" and a new pair, traveling at right angles to the original one, is "created" (fig. 4a). In all other cases, i.e., when two particles cross one another's paths at right angles (fig. 4b) or when more than two particles meet, all particles just continue straight on their paths.

#11 The definition of partitioning cellular automata introduces a minor departure from uniformity in space and time; but the full uniformity of an *ordinary* cellular automaton can easily be restored; in the present example, one would consider "super-cells" (each consisting of the four cells that make up a region of partition P_0) and "super-steps" (consisting of the composition of two consecutive steps) – yielding a global map of the form $\tau = \tau_1\tau_0$. Quite generally, a wide class of finitary rules that have a periodic spacetime structure can be recast as ordinary cellular automata.

#12 The "firing-squad" problem, of which this is a special case, dates back to the origins of cellular automata [52].

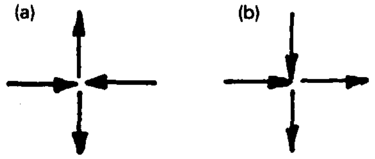

Fig. 4. In the HPP gas, particles colliding head-on are scattered at right angles (a), while particles crossing one another's paths go through unaffected (b).

As soon as the numbers involved become large enough for averages to be meaningful – say, averages over spacetime volume elements containing thousands of particles and involving thousands of collisions – a definite continuum dynamics emerges. And, in the present example, it is a rudimentary *fluid* dynamics, with quantities recognizably playing the roles of density, pressure, flow velocity, viscosity, speed of sound, etc. Fig. 5 illustrates sound-wave propagation in this model. Note that, even though the microscopic interactions only display a more limited form of rotational symmetry (namely, invariance for quarter-turn ro-

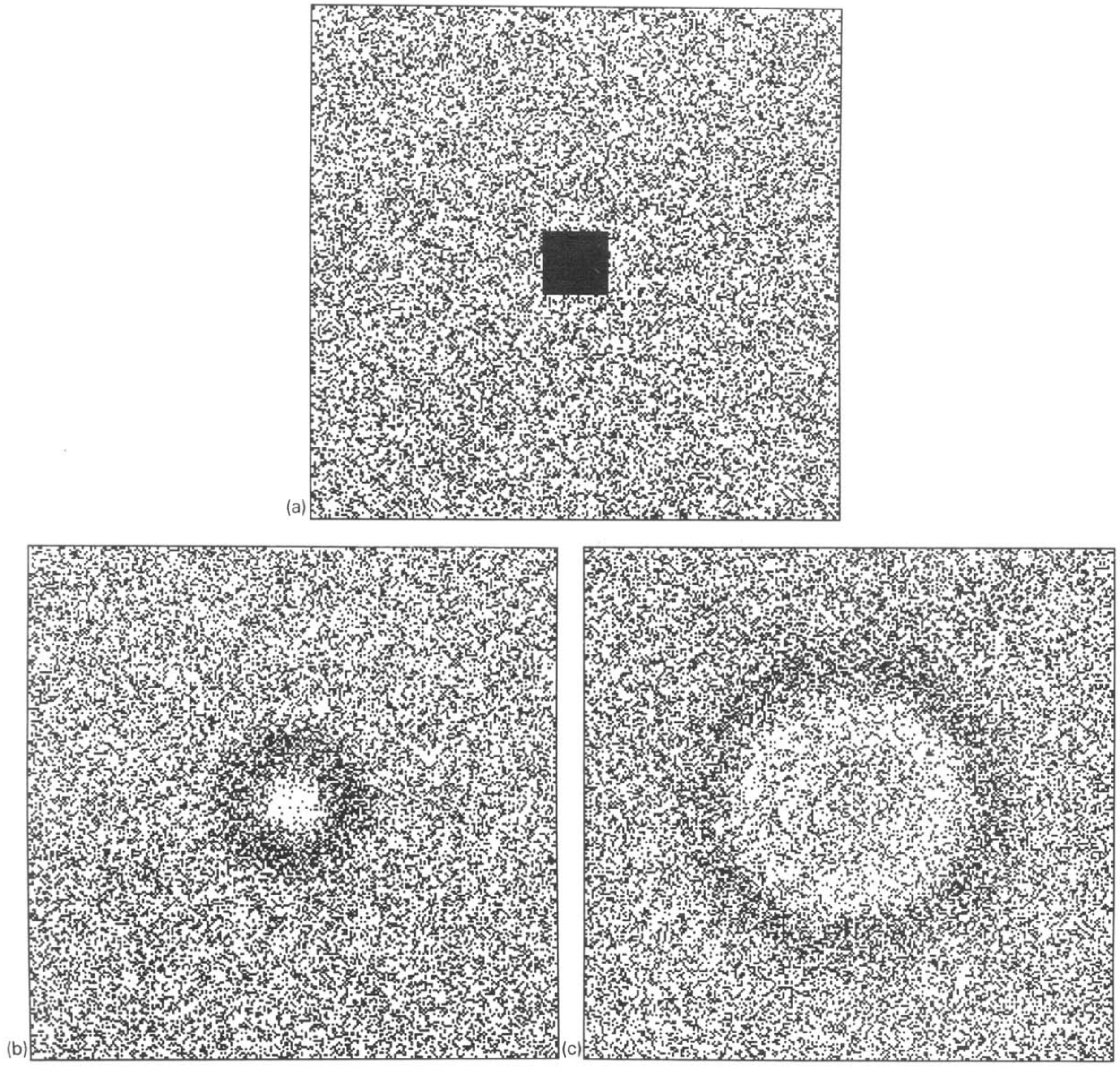

Fig. 5. Wave propagation in the HPP lattice gas. Note the emergence of circular symmetry.

tations), the speed of sound is fully isotropic.

Unlike sound speed, sound attenuation is *not* isotropic in the HPP model. It turns out that, besides conserving energy and momentum (see section 6.1), HPP separately conserves the horizontal component of momentum on each horizontal row and the vertical component on each vertical column. These *spurious* conservations (they have no counterpart in ordinary physics) lead to significant departures from the behavior one would expect from a physical fluid.

The slightly more complicated FHP lattice gas model [21] – which uses six rather than four particle directions (always in two dimensions) – gives, in an appropriate macroscopic limit, a fluid obeying the well-known *Navier–Stokes* equation, and which is thus suitable for modeling actual hydrodynamics (see ref. [27] for a tutorial). Recently, analogous results for three-dimensional models have been obtained by a number of researchers [22].

Lattice gases are rapidly beginning to encroach into modeling niches dominated until recently by differential equations [16,49]; their success in this role is chiefly due to the ease with which they can be made to satisfy local continuity equations, #13 as discussed below and, more leisurely, in refs. [72,75]).

Let us consider the spacetime texture induced by a lattice gas such as HPP. #14 In fig. 6, the arcs represent the spacetime paths available to the particles (a, b, c, d denote the four possible directions of motion) and the nodes (labeled f) represent the available collision sites; the entire structure is iterated in space and time, yielding a body-centered cubic lattice. Thus, we have a spacetime diagram (the arcs are *signals* and the nodes *events*) of a kind that is routinely used in illustrating special-relativity arguments. From this diagram, a particular history is obtained by assigning, as initial conditions, a definite occupation state ('particle' or 'no particle', which may be denoted by '0' and '1' respectively) to each arc that crosses a given spacelike surface, and then extending the state assignment timewards, using the following table at each event

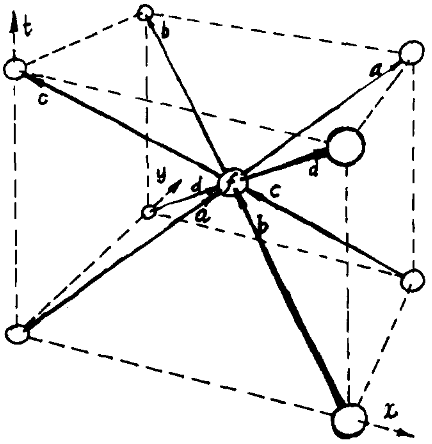

Fig. 6. Spacetime layout of the HPP gas.

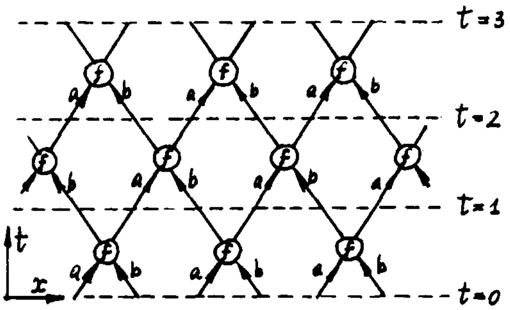

Fig. 7. Evenly spaced spacelike surfaces of the form t=constant in a lattice-gas spacetime diagram.

	in *abcd*	out *abcd*		in *abcd*	out *abcd*
	0000	0000		1000	1000
	0001	0001		1001	1001
	0010	0010	*	1010	0101
	0011	0011		1011	1011
	0100	0100		1100	1100
*	0101	1010		1101	1101
	0110	0110		1110	1110
	0111	0111		1111	1111

(18)

The table itself represents the collision rule given in words above. Note that in only two cases (marked with an asterisk) out of sixteen does an

#13 In this traditional but unfortunate term, 'continuity' does not refer to the continuum, in opposition to 'discreteness'; rather, it refers to "continuity of existence" – detailed balance, in other words, as in Kirkhhoff's laws.

#14 Much as in the case of Q2R (cf. footnote #10), also in HPP the lattice splits into two intermeshed but independent sublattices. Here we shall consider only one of these sublattices.

interaction take place; in all other cases, each of the four signals proceeds undisturbed.

Since the function f represented by this table is invertible, the spacetime history may be extended backwards as well as forwards in time.

If one now considers surfaces of the form $t =$ constant, drawn midway between rows of events, as in fig. 7 (where, for clarity, only one spatial dimension is indicated), it is clear that the collective state of the signals that traverse each such surface can be thought of as the configuration of a cellular automaton. If one groups together the four signals that converge on the same event (fig. 4), and notes the regular alternation, at odd and even times, of these groupings (cf. fig. 7), it will be evident (cf. fig. 3) that the updating scheme of fig. 4 is isomorphic to that described in section 5.5.

In other words, lattice gases and partitioning cellular automata are formally the same thing. Nonetheless, the tendency is to reserve the term 'lattice gas' for cellular automata in which the 'gas' metaphor can be defended [33]. Typically, one has a distinguished "vacuum" state ('0' in table (18)) on whose background isolated "particles" ('1' in table (18)) travel with inertial motion.

Figs. 6 and 7 make it clear not only that (a) the global dynamics is invertible if f is invertible, but also that (b) any additive quantity carried by individual signals (occupation number, momentum, etc.) is globally conserved if it is conserved by f. For example, in ref. [9] we show how certain second-order ICA can be rewritten in a straightforward way as lattice-gas ICA; conserved quantities (such as bond energy) that in the second-order format it took some effort to discover [58] are immediately visible in the lattice-gas format.

6. Physical modeling

The question that we are most often asked about cellular automata is the following.

"I've been shown cellular automata that make surprisingly good models of, say, hydrodynamics, heat conduction, wave scattering, flow through porous media, nucleation, dendritic growth, phase separation, etc. But I'm left with the impression that these are all *ad hoc* models, arrived at by some sort of magic."

"I'm a scientist, not a magician. Are there well-established *correspondence rules* that I can use to translate features of the system I want to model into specifications for an adequate cellular-automaton model of it?"

Physical modeling with cellular automata is a young discipline. Similar questions were certainly asked when differential equations were new – and, despite three centuries of accumulated experience, modeling with differential equations still requires a bit of magic. Mainly, we get new models by dressing up old models, or by microdynamical analogy. As it stands, a "rational" or "analytical" mechanics of cellular automata is beginning to take shape (cf., e.g., refs. [24,25]). Ironically, the most mature aspects of this understanding concern the modeling of *continuum* phenomena. In the rest of this section, we'll try to convey at least a feeling for the issues involved.

6.1. HPP versus Navier–Stokes

How does a ridiculously simple interaction such as that of fig. 4a (represented, in the two possible spatial orientations, by the starred entries in table (18)) possibly come close to capturing the richness of the Navier–Stokes equation?

Let us attempt to sketch an answer, using drastically simplified arguments.

By inspection of table (18), one concludes that (a′) the dynamics specified by this local map is *invertible*; (b′) *energy*, represented simply by particle count, is *conserved* by each event; (c′) the two components of *momentum*, obtained by separately counting (with a + or − sign depending on the sense) particles traveling in the two orthogonal directions, are similarly *conserved*; (d′) the dynamics is *rotationally invariant* for quarter-turn rotations; and (e′) there is *some coupling* between the two spatial *directions*.

By inspecting the Navier–Stokes equation, one concludes that this equation specifies that (a) the dynamics is *invertible*; (b) that *energy* and (c) *momentum* are *conserved* by contact processes (no action-at-a-distance); (d) the dynamics is *rotationally invariant* for continuous rotations; (e) pressure is a *scalar* quantity, i.e., is independent of *direction*; and (f) *nothing else* that doesn't already follow from (a)–(e).

In the cellular-automaton model, the moment one puts on *macroscopic* glasses, as it were, and is only able to see details that are much coarser than the pitch of the lattice, the discreteness of energy, momentum, and position washes out, so that points (a′), (b′), and (c′) above closely match (a), (b), and (c).

As for point (d′), it is well known that 90° rotational invariance is sufficient to yield full isotropy for *diffusive* phenomena. #15 However, in a lattice gas not too far from equilibrium (cf. ref. [73]) the chief transport mechanisms are both diffusive transport and wavelike transport, and for the full isotropy of *wavelike* phenomena it turns out that one needs nondiffusive transport of horizontal momentum in the vertical direction and vice versa – which is hard to achieve with 90° invariance and is much easier to achieve with six particle directions and 60° invariance (cf., e.g., ref. [88]). This is why the fluid modeled by the HPP gas significantly departs from the Navier–Stokes equation (cf. section 5.6), and why a model such as FHP manages to achieve the goal.

As for point (e′), by considerations similar to the above one can show that *almost any* coupling between the x and y directions leads to equalization of pressure in all directions.

The only philosophical difficulty lies with the 'nothing else' of point (f). Like sorcerer's apprentices, we commanded HPP to conserve a certain few quantities. HPP complied; but it didn't warn us that it had taken the initiative to conserve an infinity of other quantities as well (section 5.6). And these spurious conservations undermined our modeling efforts.

How do we know that something like this will not happen to us the next time? How can one tell that a model has no spurious invariants [15]? (For that matter, how does one know that a certain unanticipated invariance is 'spurious'? may it not turn out to be a "feature" rather than a "bug"? #16) A general method for keeping spurious constraints out of a cellular-automaton model is suggested by Jaynes's maximum entropy principle [36]. The idea is to make the local map *maximally random* with respect to any features that are not explicitly demanded by the modeling context. #17 Unfortunately, to guarantee a closer and closer approximation to this ideal of maximal randomness one may have to introduce larger and larger state alphabets and neighborhoods, at a corresponding sacrifice in simulation efficiency. In the end, one must know where to draw the line between accuracy and efficiency.

6.2. Invertibility, symmetries, and conserved quantities

An invertible cellular automaton shares certain important traits with physical systems even when it does not manifestly support waves, particles, energy, momentum, and similar symptoms of "physicalness".

Even in dynamical systems that have very little structure one can show some connections between symmetries and conservation laws. #18 However, such connections manifest themselves with special strength in Hamiltonian systems, where, according to Noether's theorem, to each continuous one-parameter group of transformations that commutes with the dynamics there is associated a real-valued, conserved quantity. Thus, energy is associated with time-invariance of the dynamics, the three components of momentum with translation invariance, etc. These quantities are functions of

#15 Consider the set of all possible algorithms telling an individual how to move, one block at a time, North, South, East, or West on an orthogonal street grid (at each step, the choice may take into account local landscape features, the actions of neighboring individuals, the weather, etc.). *Almost all* such algorithms yield an exploration pattern that is close to a superposition of independent random walks in the two orthogonal directions. In the limit as the binomial distribution goes over to the Gaussian distribution, such a pattern displays full rotational invariance (i.e., $\exp(x^2)\exp(y^2) = \exp(x^2+y^2) = \exp(r^2)$) – and this in spite of the fact that the microscopic rule can at best have quarter-turn invariance.

#16 For instance, certain constraints that in lattice gases lead to a departure from Galilean invariance [28] may actually bias the system toward *Lorentz* invariance [75] (also cf. ref. [65]).

#17 Hénon's strategy for rule optimization in lattice gas hydrodynamics [32] is somewhat related to this approach.

#18 Given any group of transformations that commutes with the dynamics, each configuration belongs to a definite symmetry class (a normal subgroup of the group itself); all the points of an orbit belong to the same symmetry class, which is thus a *conserved quantity*.

the system's state as such (i.e., its "mechanical" or "microscopic" state); one can literally speak of, say, the *energy content* of a certain state.

The Hamiltonian structure itself, i.e., invariance with respect to canonical transformations, leads to the conservation of an additional quantity, called *(fine-grained) entropy*, which is defined not on individual mechanical states but on *statistical* states (probability distributions on the set of mechanical states); a special case of this conservation is the well-known "incompressibility" of volume in phase space (Liouville's theorem). In other words, (a) the Hamiltonian structure provides an unequivocal way of measuring the *information* content [17, ch. II] of a statistical state, and (b) the quantity thus measured is *conserved* by the dynamics.

Continuum dynamical systems that do not have a Hamiltonian structure lack, in general, an intrinsic "yardstick" for measuring the information content of a (statistical) state, and thus do not come with a built-in statistical mechanics. Cellular automata, on the other hand, owing to their local finiteness (the lattice is discrete, and state alphabet and neighborhood are finite), do possess a natural information measure. #19 And the foremost property of *invertible* cellular automata is, of course, that they are *information-conserving.*

Because of this "information-losslessness" (ach!), ICA automatically obey the second principle of thermodynamics and, more generally, display a full-featured statistical mechanics analogous to that of Hamiltonian systems. As additional structure is introduced (for instance, particle conservation), macroscopic mechanical features such as elasticity, inertia, etc. naturally emerge out of statistics itself. In sum, once we make sure that it is conserved, information has an irresistible tendency to take on a strikingly tangible aspect (cf. ref. [73]) – to *materialize* itself.

For the above reasons, and because they lend themselves to very efficient computer simulations, ICA are an ideal medium for the qualitative study of the connections between microscopic mechanics and statistical mechanics on one hand (cf. refs. [43,77,85,13]), and between statistical mechanics and macroscopic mechanics on the other (cf. ref. [88]). They are also suitable for the the modeling of an increasingly important type of generalized mechanical activity, namely *computation* [43].

In physics, additive invariants, whether represented by mechanical quantities such as energy or statistical quantities such as entropy, bear a major responsibility for the emergence of nontrivial macroscopic properties. Intuitively, one may expect that almost every detail of the microscopic interactions will be washed out by macroscopic averaging; only features that are supported by a definite conspiracy (such as a particular symmetry or conservation law) will bubble up all the way to the macroscopic surface and emerge there as recognizable laws. The study of invariants in ICA has barely started. Here we can only refer the reader to a few literature items, such as refs. [58,43,15,66]. It must be noted that in general the problem of *discovering* the invariants of a cellular automaton is presumably of a difficulty comparable to that of deciding its invertibility (cf. section 4). Much as for invertibility, it is much easier to directly *synthesize* a cellular automaton having certain desired invariants than to figure out the invariants of a given one.

6.3. Modeling first principles?

As we hinted in sections 5.6–6.2, ICA are quite successful at explaining complex macroscopic phenomenology in terms of the collective behavior of an enormous number of very simple subsystems. So much so, that one begins to wonder whether aspects of physics that are usually regarded as primitive (such as the principles of analytical mechanics) are not, after all, similarly derivable as emergent aspects of an extremely fine-grained underlying dynamical structure.

In this role, ICA have been used with some success as fine-grained models of basic aspects of special and general relativity [75,80,65] and of quantum mechanics [34]

7. Decidability, revisited

Now that we have some concrete examples of ICA in mind, let us pick up the trail that we left off at the end of section 4.4.

#19 This is the *uniform* measure, which gives equal weight to all configurations.

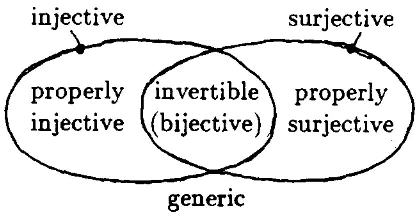

Fig. 8. Venn diagram for surjectivity and injectivity.

cellular automata:

invertible (r.e)	properly surjective	nonsurjective (r.e.)

Fig. 9. Surjectivity and injectivity in cellular automata; 'r.e.' denotes a recursively enumerable set.

A dynamical system is *surjective* if every state has at least one predecessor; *injective*, if it has at most one predecessor. If it is both surjective and injective it is, of course, invertible (or *bijective*). It will be convenient to call *properly surjective* a surjective system that is not invertible, and similarly *properly injective* an injective system that is not invertible. The situation is summed up in fig. 8.

For cellular automata, the 'properly injective' lobe in fig. 8 is empty, i.e., injectivity is equivalent to invertibility (Richardson [60]), leaving us with the simpler situation of fig. 9.

As we have seen in section 4.4, the left set of fig. 9 (i.e., the class of invertible cellular automata) is recursively enumerable (theorem 4.3). The right set too is recursively enumerable, according to the following

Theorem 7.1 *The class of nonsurjective cellular automata is recursively enumerable.*

Proof. According to Myhill's theorem [53] (cf. section 2.3), if a configuration has no predecessors there must exist a configuration having two predecessors that are identical except over a finite area. If such a pair exists, brute-force enumeration (apply the local map to larger and larger finite areas) will eventually turn it up. □

If also the middle set in fig. 9, i.e., the set of properly surjective (PS) cellular automata, were recursively enumerable, then all three sets would be fully *recursive* (i.e., the corresponding predicates would be decidable). From theorem 4.5, we must conclude that the middle set is not even recursively enumerable. We shall examine the makeup of this PS set in more detail.

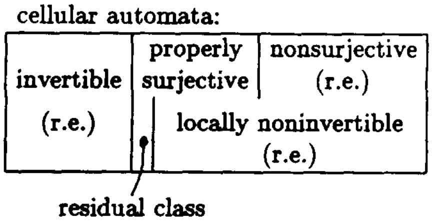

Fig. 10. Between invertibility and local noninvertibility, which are both semidecidable, there is a no man's land that is not semidecidable.

Let us apply the local map of a cellular automaton on a *finite*, wrapped-around space array (which is equivalent to imposing periodic boundary conditions along each dimension). #20 The resulting *finite cellular automaton* will be invertible if the original was invertible, and nonsurjective if the original was nonsurjective. If, however, the original was PS, then the resulting cellular automaton will be forced to abandon 'PS-ness' and "choose" between being invertible and being nonsurjective (by a simple counting argument, the PS middle ground is forbidden); note that either choice may occur with the *same* local map, for different sizes of the finite space.

Those cellular automata that become nonsurjective when thus forced on a finite space (at least for some space size), together with those that were nonsurjective to begin with, will be called *locally noninvertible* (they are exactly those cellular automata whose noninvertibility can be verified by local arguments). Local noninvertibility is semidecidable (the proof is similar to that of theorem 7.1). What is left (fig. 10) is a residual class of cellular automata having rather counterintuitive behavior; that is, they

– are invertible on *all* finite spaces, but
– become noninvertible (specifically, properly surjective) on the infinite array.

Because of theorem 4.5, this residual class cannot be empty; on the other hand, no instance of

#20 To simplify the statement of some of the assertions below, we shall consider only finite spaces that are large enough to keep all of the neighbors of a cell distinct.

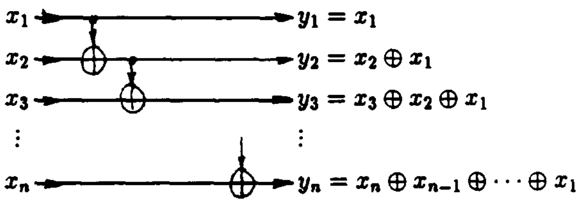

Fig. 11. An invertible function with an asymmetric dependency pattern.

this class has, to our knowledge, ever been exhibited.

Let us go back to the enumeration in the proof of theorem 4.3. If from λ we could determine an upper bound to the neighborhood radius of its hypothetical inverse, then we would, eventually, positively know whether or not this inverse exists. From theorem 4.5, we must conclude that this is not the case; in particular, for any n there must be ICA for which the radius of the *inverse neighborhood* (the neighborhood of the inverse) is at least n times larger than the radius of the *direct* neighborhood.

We shall show how to construct such an ICA, for any n (note that in this example n, though arbitrarily large, is a known function of the size of the state alphabet, so that we *do* have an upper bound for the neighborhood radius). Consider the function with n binary inputs $x_1, \ldots, x_n$ and n binary outputs $y_1, \ldots, y_n$ defined by

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= x_2 \oplus x_1 && (= x_2 \oplus y_1), \\ y_3 &= x_3 \oplus x_2 \oplus x_1 && (= x_3 \oplus y_2), \\ &\ldots \\ y_n &= x_n \oplus x_{n-1} \oplus \cdots \oplus x_1 && (= x_n \oplus y_{n-1}), \end{aligned} \tag{19}$$

where $\oplus$ denotes Boolean exclusive-OR. This function, pictorially shown in fig. 11, is invertible, and its inverse is given by

$$\begin{aligned} x_1 &= y_1, \\ x_2 &= y_2 \oplus y_1, \\ x_3 &= y_3 \oplus y_2, \\ &\ldots \\ x_n &= y_n \oplus y_{n-1}. \end{aligned} \tag{20}$$

Note the asymmetric dependency pattern: a y may depend on up to n of the x's, while each x depends on no more than *two* of the y's.

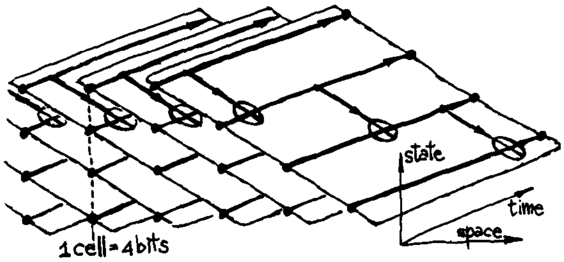

Fig. 12. Infinite juxtaposition of copies of function (19) yields a one-dimensional ICA having n bits per cell. Here $n = 4$ for definiteness.

From (19), by using a step-and-repeat construction (fig. 12), we obtain a one-dimensional cellular automaton having n bits per cell. This cellular automaton is invertible; going forward in time, the new state of each cell depends on the current state of itself and the $n-1$ cells to its left; going backwards, each cell depends only on itself and the cell on its left. The neighborhood radii for the direct local map λ and the inverse local map $\overline{\lambda}$ are, respectively, n and 1.

This cellular automaton is an invertible system governed by local and uniform laws, but the cause-and-effect tree (the "light cone") in one direction of time is much wider than in the opposite direction.[#21] Such behavior doesn't seem to have any counterpart in physics.

8. Structural invertibility

The usual definition of a cellular automaton, as given in section 2.1, is a *structural* one; i.e., one explicitly tells how a cellular automaton *is made.*[#22] The structure in question is a *uniform sequential*

[#21] See ref. [57] for a related discussion on different aspects of causality.

[#22] One can think of the overall state of the system as a point in some abstract space, and of a configuration as a way of expressing this point in a convenient coordinate system (each site represents a coordinate); the local map tells how to update the state-component associated with each coordinate. In section 2.4 we mentioned the possibility of an equivalent *functional* definition, where the global map is characterized in terms of what it *does* to the state points themselves, without making recourse to coordinates.

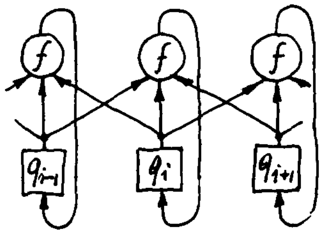

Fig. 13. A cellular automaton as a sequential network. We illustrate the case of one dimension and neighborhood $X = \langle -1, 0, +1 \rangle$. The table is denoted by f.

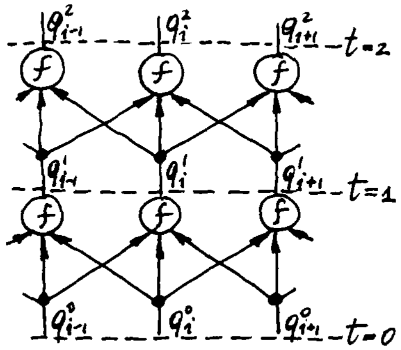

Fig. 14. The cellular automaton of fig. 13, re-expressed as a combinational network. The f nodes denote occurrences of the table; the black dots, occurrences of fan-out.

network: the state q_i of a cell (a square in fig. 13) is realized by a collection of flip-flops or similar memory elements, and the table f (a circle in the same figure) by a composition of NAND gates or similar logic elements. Both kinds of element have a straightforward implementation in a number of technologies.

Since a flip-flop is simply a digital sample-and-hold device, used for regulating the traffic in and out of the gates, #23 it will be conceptually simpler to represent the design of a cellular automaton as a uniform *combinational* network, where each usage of a table is explicitly represented by a separate node, as in fig. 14a.

Now, if we were given the design of a deterministic physical machine A whose behavior happens to be invertible (i.e., for any possible state there is a unique way one can arrive at it), we would tend to think that

(i) One can construct, out of the same technology, a machine $\overline{A}$ that has the same orbit structure as A but follows each orbit in reverse.

(ii) From the detailed design plans of machine A one should be able to arrive in a straightforward way at the design plans for machine $\overline{A}$.

Both points are certainly true if the "technology" in question is Newtonian mechanics (think of the sun and the planets). In fact, since this mechanics is time-reversal invariant (cf. footnote #2), we can use the *same* machinery ($\overline{A} = A$), and just reverse the direction of each particle in order to traverse orbits backwards.

Lemma 4.1 tells us that point (i) above is true also for more practical technologies, where non-Newtonian devices such as amplifiers, latches, dampers, and other dissipative devices are available. Note that the usual way cellular automata are specified tacitly implies the availability of such a dissipative technology. In fact, the f nodes in fig. 13 are typically noninvertible because they are *noninjective*, and the fan-out nodes noninvertible because they are *nonsurjective*; as explained in [19], nonsurjective computing primitives imply the availability of a *power supply*, and noninjective ones imply a *heat sink*.

However, theorem 4.5 says that, in this dissipative context, point (ii) is no longer tenable. That is, from design plans of the type represented by fig. 14 we do not know, in general, how to make similar design plans for a new machine that, if the original machine was invertible, will have exactly the inverse behavior. Thus, to rescue point (ii) we have to turn our attention to design plans that do not imply dissipative primitives. For *finite* invertible automata this is always possible [70,19] (see also [69] for the analogous problem in continuous systems); is it possible for cellular automata?

Now that the job of motivating it is done, let us reword the above question. Given an arbitrary ICA, which can always be thought of as realized by a uniform combinational network of the type of fig. 14, is it always possible to give an alternative realization of it by means of a uniform combinational network in which (a) all nodes stand for *invertible* functions, and (b) no extra state variables are introduced? We shall call such a realization

#23 Each gate is used over and over, at each time step evaluating the same function on a new set of arguments.

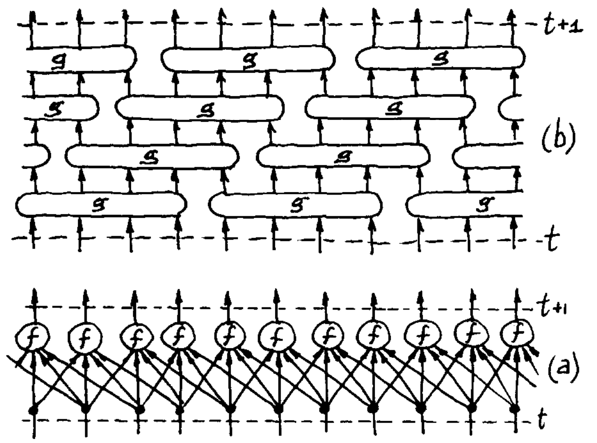

Fig. 15. (a) Combinational network representation of the ICA defined by table (9). (b) The same ICA, in a version that exhibits structural invertibility; the nodes are all invertible Boolean functions.

structurally invertible;[#24] we shall also call *structurally invertible* an ICA for which a structurally invertible realization exists.

It is clear that from a structurally invertible realization one immediately obtain design plans for the inverse automaton.

We have succeeded in exhibiting structural invertible realizations for every ICA for which we tried (here we shall give only one example, in fig. 15), and apparently all ICA we are aware of are structurally invertible. From the above considerations, we are led to the following

Conjecture 8.1 *All invertible cellular automata are* structurally *invertible, i.e., can be (isomorphically) expressed in spacetime as a uniform composition of finite invertible logic primitives.*

In other words, we conjecture that there are no cellular automata for which invertibility (a *functional* property) cannot be explained in terms of structural invertibility, which is, of course, a *structural* property.

For an example, fig. 15 shows a structurally invertible design for the conserved-landscape ICA discussed in section 5.3, and contrasts it with the conventional design. The function f is given by table (9); g is an invertible Boolean function with four inputs and four outputs obtained in a straighforward way from f as follows:

$$\begin{aligned} y_1 &= x_1, \\ y_2 &= f(x_1, x_2, x_3, x_4) = (\overline{x}_1 \cdot x_3 \cdot \overline{x}_4) \oplus x_2, \\ y_3 &= x_3, \\ y_4 &= x_4, \end{aligned} \tag{21}$$

where a dot denotes Boolean AND, a bar denotes Boolean complement, and the order (1, 2, 3, 4) of inputs and outputs corresponds to left-to-right in fig. 15.

Note that, even though the new network is uniform, its spatial "pitch" is coarser than that of the original network; in other words, in fig. 15b we have a structure whose *function* is left invariant by shifting, say, one position to the left, but whose *structure* needs to be shifted *four* positions before it again coincides with itself![#25] We conjecture that, for a structurally invertible realization, recourse to a network having a *coarser* pitch than the original one is, in general, unavoidable.

9. Conclusions

We have presented invertible cellular automata from a number of angles, trying to illustrate the concrete motivations of many theoretical questions.

This paper also constitutes an original contribution to the study of the mechanisms and pathways of *causality* in distributed reversible systems.

Acknowledgements

This research was supported in part by the Defense Advanced Research Projects Agency (N00014-89-J-1988), and in part by the National Science Foundation (8618002-IRI).

[#24] Condition (b) simply asks for an *isomorphic* realization. Realizations that satisfy (a) but make use of extra state variables can be obtained by a straightforward procedure, as shown in ref. [35].

[#25] More formally, the uniformity group of the structural description is a proper subgroup of that of the functional behavior.

References

[1] V. Aladyev, Computability in homogeneous structures, Izv. Akad. Nauk. Estonian SSR, Fiz.-Mat. 21 (1972) 80–83.

[2] S. Amoroso and Y. N. Patt, Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures, J. Comp. Syst. Sci. 6 (1972) 448–464.

[3] S. Amoroso et al., Some clarifications of the concept of a Garden-of-Eden configuration, J. Comp. Syst. Sci. 10 (1975) 77–82.

[4] E. Banks, Information processing and transmission in cellular automata, Tech. Rep. MAC TR-81, MIT Project MAC (1971).

[5] P. Benioff, Quantum mechanical Hamiltonian models of discrete processes that erase their own histories: applications to Turing machines, Int. J. Theor. Phys. 21 (1982) 177–201.

[6] C. Bennett, Logical reversibility of computation, IBM J. Res. Develop. 6 (1973) 525–532.

[7] C. Bennett and G. Grinstein, Role of irreversibility in stabilizing complex and nonenergodic behavior in locally interacting discrete systems, Phys. Rev. Lett. 55 (1985) 657–660.

[8] C. Bennett, On the nature and origin of complexity in discrete, homogeneous, locally-interacting systems, Found. Phys. 16 (1986) 585–592.

[9] C. Bennett, N. Margolus and T. Toffoli, Bond-energy variables for Ising spin-glass dynamics, Phys. Rev. B 37 (1988) 2254.

[10] C. Bennett, T. Toffoli and S. Wolfram, eds., Cellular Automata '86, Tech. Memo MIT/LCS/TM-317, MIT Lab. for Comp. Sci. (December 1986).

[11] A. Burks, ed., Essays on Cellular Automata (University of Illinois Press, Champaign, IL, 1970).

[12] R. Courant, et al., On the partial difference equations of mathematical physics, Math. Ann. 100 (1928) 32–74 (in German), republished in English translation in IBM J. (March 1967) 215–238.

[13] M. Creutz, Deterministic Ising dynamics, Ann. Phys. 167 (1986) 62–76.

[14] K. Culik, On invertible cellular automata, Complex Systems 1 (1987) 1035–1044.

[15] Invariants in lattice gas models, in: Discrete Kinetic Theory, Lattice Gas Dynamics, and Foundations of Hydrodynamics, ed. R. Monaco (World Scientific, Singapore, 1989) pp. 102–113.

[16] G. Doolen et al., eds., Lattice-Gas Methods for Partial Differential Equations (Addison–Wesley, New York, 1990).

[17] H. Everett, The theory of the universal wave function, in: The Many-World Interpretation of Quantum Mechanics, eds. B. DeWitt and N. Graham (Princeton Univ. Press, Princeton, NJ, 1973).

[18] R. Feynman, Quantum-mechanical computers, Opt. News 11 (Feb. 1985) 11–20, reprinted in Found. Phys. 16 (1986) 507–531.

[19] E. Fredkin and T. Toffoli, Conservative logic, Int. J. Theor. Phys. 21 (1982) 219–253.

[20] E. Fredkin, R. Landauer and T. Toffoli, eds., Proceedings of a Conference on Physics and Computation, Int. J. Theor. Phys. 21(3/4), 21(6/7), 21(12) (1982).

[21] U. Frisch, B. Hasslacher and Y. Pomeau, Lattice-gas automata for the Navier–Stokes equation, Phys. Rev. Lett. 56 (1986) 1505–1508.

[22] U. Frisch et al., Lattice gas hydrodynamics in two and three dimensions, in: Lattice-Gas Methods for Partial Differential Equations, eds. G. Doolen et al. (Addison–Wesley, New York, 1990) pp. 77–135.

[23] M. Gardner, The fantastic combinations of John Conway's new solitaire game 'Life', Sci. Am. 223 (April 1970) 120–123.

[24] H. Gutowitz, J.D. Victor and B.W. Knight, Local structure theory for cellular automata, Physica D 28 (1987) 18–48.

[25] H.A. Gutowitz, A hierarchical classification of cellular automata, Physica D 45 (1990) 136–156, these Proceedings.

[26] J. Hardy, O. de Pazzis and Y. Pomeau, Molecular dynamics of a classical lattice gas: transport properties and time correlation functions, Phys. Rev. A 13 (1976) 1949–1960.

[27] B. Hasslacher, Discrete fluids, Los Alamos Science, Special Issue No. 15 (1987) 175–200, 211–217.

[28] F. Hayot, The effect of Galilean non-invariance in lattice gas automaton one-dimensional flow, Complex Systems 1 (1987) 753–761.

[29] T. Head, One-dimensional cellular automata: injectivity from unambiguity, to appear in Complex Systems 3 (1989).

[30] G.A. Hedlund, K. I. Appel and L. R. Welch, All onto functions of span less than or equal to five, Communications Research Division, working paper (July 1963).

[31] G.A. Hedlund, Endomorphism and automorphism of the shift dynamical system, Math. Syst. Theory 3 (1969) 51–59.

[32] M. Hénon, Optimization of collision rules in the FCHC lattice gases and addition of rest particles, in: Discrete Kinetic Theory, Lattice Gas Dynamics, and Foundations of Hydrodynamics, ed. R. Monaco (World Scientific, Singapore, 1989) pp. 146–159.

[33] M. Hénon, On the relation between lattice gases and cellular automata, in: Discrete Kinetic Theory, Lattice Gas Dynamics, and Foundations of Hydrodynamics, ed. R. Monaco (World Scientific, Singapore, 1989) pp. 160–161.

[34] H. Hrgovčić, personal communication.

[35] G. Jacopini and G. Sontacchi, Reversible parallel computation: an evolving space model, to appear in Theor. Comp. Sci. 75 (1990).

[36] Jaynes, Information theory and statistical mechanics, Phys. Rev. 106 (1957) 620–630; 108 (1957) 171–190.

[37] K. Kaneko, Symplectic cellular automata, Phys. Lett. A 129 (1988) 9–16.

[38] J. Kari, Reversibility and surjectivity of cellular au-

tomata, Licentiate's Thesis, University of Turku, Finland (May 1989).

[39] J. Kari, Reversibility of 2D cellular automata is undecidable, Physica D 45 (1990) 379–385, these Proceedings.

[40] A. Lindenmayer and G. Rozenberg, eds., Automata, Language, and Development (North-Holland, Amsterdam, 1976).

[41] N. Margolus, Physics-like models of computation, Physica D 10 (1984) 81–95.

[42] N. Margolus, Quantum computation, Ann. NY Acad. Sci. 480 (1986) 487–497.

[43] N. Margolus, Physics and Computation, Ph. D. Thesis, Tech. Rep. MIT/LCS/TR-415, MIT Lab. for Comp. Sci. (March 1988).

[44] N. Margolus and T. Toffoli, Cellular Automata Machines, in: Lattice-Gas Methods for Partial Differential Equations, eds. G. Doolen et al. (Addison–Wesley, New York, 1990) pp. 219–248.

[45] N. Margolus, T. Toffoli and G. Vichniac, Cellular-automata supercomputers for fluid dynamics modeling, Phys. Rev. Lett. 56 (1986) 1694–1696.

[46] A. Maruoka and M. Kimura, Conditions for injectivity of global maps for tessellation automata, Info. Control 32 (1976) 158–162.

[47] A. Maruoka and M. Kimura, Injectivity and surjectivity of parallel maps for cellular automata, J. Comp. Syst. Sci. 18 (1979) 47–64.

[48] A. Maruoka and M. Kimura, Strong surjectivity is equivalent to C-injectivity, Theor. Comp. Sci. 18 (1982) 269–277.

[49] R. Monaco, ed., Discrete Kinetic Theory, Lattice Gas Dynamics, and Foundations of Hydrodynamics (World Scientific, Singapore, 1989).

[50] K. Morita and M. Harao, Computation universality of one-dimensional reversible (injective) cellular automata, Trans. IEICE E 72 (1989) 758–762.

[51] E. Moore, Machine models of self-reproduction, Proc. Symp. Appl. Math. 14 (1962) 17–33; reprinted in: Essays on Cellular Automata, ed. A. Burks (University of Illinois Press, Champaign, IL, 1970) pp. 187–203.

[52] E. Moore, The firing squad synchronization problem, in: Sequential Machines, ed. E. F. Moore (Addison–Wesley, New York, 1964) pp. 213–214.

[53] J. Myhill, The converse of Moore's garden-of-Eden theorem, Proc. Am. Math. Soc. 14 (1963) 658–686; reprinted in: Essays on Cellular Automata, ed. A. Burks (University of Illinois Press, Champaign, IL, 1970) pp. 204–205.

[54] M. Nasu, Local maps inducing surjective global maps of one-dimensional tessellation automata, Math. Syst. Theor. 11 (1978) 327–351.

[55] N. Packard and S. Wolfram, Two-dimensional cellular automata, J. Stat. Phys. 38 (1985) 901–946.

[56] Y.N. Patt, Injections of neighborhood size three and four on the set of configurations from the infinite one-dimensional tessellation automata of two-state cells (unpublished report, cited in ref. [2]), ECON-N1-P-1, Ft. Monmouth, NJ 07703 (1971).

[57] C.A. Petri, State-transition structures in physics and in computation, in: Proceedings of a Conference on Physics and Computation, eds. E. Fredkin, R. Landauer and T. Toffoli, Int. J. Theor. Phys. 21 (1982) 979–992.

[58] Y. Pomeau, Invariant in cellular automata, J. Phys. A 17 (1984) L415–L418.

[59] K. Preston and M. Duff, Modern Cellular Automata (Plenum Press, New York, 1984).

[60] D. Richardson, Tessellation with local transformations, J. Comp. Syst. Sci. 6 (1972) 373–388.

[61] T. Sato and N. Nonda, Certain relations between properties of maps of tessellation automata, J. Comp. Syst. Sci. 15 (1977) 121–145.

[62] M. Sears, The automorphisms of the shift dynamical systems are relatively sparse, Math. Syst. Theory 5 (1971) 228–231.

[63] A. Smith, Cellular automata theory, Tech. Rep. 2, Stanford Electronic Lab., Stanford Univ. (1969).

[64] S. Takesue, Reversible cellular automata and statistical mechanics, Phys. Rev. Lett. 59 (1987) 2499–2502.

[65] M.A. Smith, Representations of geometrical and topological quantities in cellular automata, Physica D 45 (1990) 271–277, these Proceedings.

[66] S. Takesue, Relaxation properties of elementary reversible cellular automata, Physica D 45 (1990) 278–284, these Proceedings.

[67] T. Toffoli, Computation and construction universality of reversible cellular automata, J. Comp. Syst. Sci. 15 (1977) 213–231.

[68] T. Toffoli, Cellular automata mechanics, Tech. Rep. 208, Comp. Comm. Sci. Dept., The University of Michigan (1977).

[69] T. Toffoli, Bicontinuous extension of reversible combinatorial functions, Math. Syst. Theory 14 (1981) 13–23.

[70] T. Toffoli, in: Reversible Computing, Automata, Languages and Programming, eds. de Bakker and van Leeuwen (Springer, Berlin, 1980) pp. 632–644.

[71] T. Toffoli, CAM: A high-performance cellular-automaton machine, Physica D 10 (1984) 195–204.

[72] T. Toffoli, Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics, Physica D 10 (1984) 117–127.

[73] T. Toffoli, Information transport obeying the continuity equation, IBM J. Res. Develop. 32(1) (January 1988) 29–36.

[74] T. Toffoli, Four topics in lattice gases: ergodicity; relativity; information flow; and rule compression for parallel lattice-gas machines, in: Discrete Kinetic Theory, Lattice Gas Dynamics, and Foundations of Hydrodynamics, ed. R. Monaco (World Scientific, Singapore, 1989) pp. 343–354.

[75] T. Toffoli, How cheap can mechanics' first principles be?, to appear in: Complexity, Entropy, and the Physics of Information, ed. W.H. Zurek (Addison–

Wesley, Reading, MA, 1990).
[76] T. Toffoli, Frontiers in computing, in: Information Processing 89, ed. G.X. Ritter (North-Holland, Amsterdam, 1989) p. 1.
[77] T. Toffoli and N. Margolus, Cellular Automata Machines – A New Environment for Modeling (MIT Press, Cambridge, MA, 1987).
[78] T. Toffoli and N. Margolus, Programmable matter, in: Lattice Gas Methods for PDEs, ed. G. Doolen, Physica D 46 (1990), special issue, to appear.
[79] T. Toffoli and N. Margolus, Invertible cellular automata, occasionaly updated technical memo (MIT Lab. for Comp. Sci.), eventually to appear in book form.
[80] G. 't Hooft, A two-dimensional model with discrete general coordinate-invariance, preprint.
[81] A. Turing, On computable numbers, with an application to the Entscheidungsproblem, Proc. London Math. Soc. Ser. 2, 42 (1936) 230–265.
[82] S. Ulam, Random processes and transformations, Proc. Int. Congr. Mathem. (held in 1950) 2 (1952) 264–275.
[83] G. Vichniac, Simulating physics with cellular automata, Physica D 10 (1984) 96–115.
[84] J. von Neumann, Theory of Self-Reproducing Automata, edited and completed by A. Burks (University of Illinois Press, Champaign, IL, 1966).
[85] S. Wolfram, Statistical mechanics of cellular automata, Rev. Mod. Phys. 55 (1983) 601.
[86] S. Wolfram, Universality and complexity in cellular automata, Physica D 10 (1984) 1–35.
[87] S. Wolfram, Computation theory of cellular automata, Commun. Math. Phys. 96 (1984) 15–57.
[88] S. Wolfram, Cellular automaton fluids 1: basic theory, J. Stat. Phys. 45 (1986) 471–526.
[89] S. Wolfram, Theory and Applications of Cellular Automata (World Scientific, Singapore, 1986).
[90] T. Yaku, Inverse and injectivity of parallel relations induced by cellular automata, Proc. Am. Math. Soc. 58 (1976) 216–220.
[91] K. Zuse, Rechnender Raum (Vieweg, Braunschweig, 1969); translated as Calculating Space, Tech. Transl. AZT-70-164-GEMIT, MIT Project MAC (1970).