International Journal of Theoretical Physics, Vol. 27, No. 7, 1988

# Intractable Problems in Reversible Cellular Automata

**Farrokh Vatan**[1]

*Received February 19, 1988*

The billiard ball model, a classical mechanical system in which all parameters are real variables, can perform all digital computations. An eight-state, 11-neighbor reversible cellular automaton (an entirely discrete system in which all parameters are integer variables) can simulate this model. One of the natural problems for this system is to determine the shape of a container so that the initial specific distribution of gas molecules eventually leads to a predetermined distribution. This problem is PSPACE-complete. Related intractable and decidable problems are discussed as well.

## 1. INTRODUCTION

It is generally accepted that reversible universal computing models reflect the basic laws of physics, and with a theory of reversible computing we can attain a connection between abstract computing and (microscopic) physical laws. It is worth considering this theory, because any concrete implementation of abstract computing is based on such laws.

In recent years, based on pioneering work of Fredkin and Toffoli (1982), it has become possible to get explicit connections between abstract computational models and physical phenomena. They have shown that the billiard ball model (BBM) can perform all digital computations, and thus it has computational universality. This model is a classical mechanical system which obeys a continuous dynamics, and all its parameters (coordinates, times, velocities, etc.) are real variables. The universality of this system has been proved by showing that it can represent the "conservative logic" gates.

Margolus (1984) has given some reversible cellular automata that duplicate the behavior of the BBM. A *cellular automaton* (CA) consists of a "space," which is divided into cubes (cells) of uniform size. Each cell can be in one of a finite number of states. The states of all cells change

[1]Department of Mathematics, Isfahan University of Technology, Isfahan, Iran.



0020-7748/88/0700-0801$06.00/0 © 1988 Plenum Publishing Corporation

simultaneously in discrete time steps. Each cell has a set of neighboring cells, and the neighborhood patterns of all cells are equal. After each time step, the state of each cell changes according to the states of its neighboring cells at a previous time. This is determined by a *local* (*transition*) *map*, which acts uniformly everywhere. The states of all cells at each time determine a *configuration* of the CA. The time evolution of the system leads to a *parallel* (*transition*) *map*, which acts on configurations. A CA is said to be *reversible* if its parallel map is bijective. [For formal definitions see, e.g., Di Gregorio and Trautter (1975), Richardson (1972), and Toffoli (1977).]

Margolus (1984) has given two nonstandard reversible cellular automata for simulating the BBM. The first is a two-state, four-neighbor automaton, but the neighboring cells of each cell are not fixed. This automaton can be defined as an eight-state, 25-neighbor standard reversible CA; but in this new automaton the "balls" jump and "collisions" occur with delay. The second is a three-state, nine-neighbor second-order automaton; i.e., the state of a cell at the time $t+1$ depends not only on the state of its neighbors at the time $t$, but also on its state at the time $t-1$. It can be defined as a nine-state, nine-neighbor standard CA; but in this new automaton each "ball" leaves its trace on its back and "mirrors" consist of a group of resting "balls."

We will show that it is possible to simulate the BBM by a standard eight-state, 11-neighbor, reversible cellular automaton.

## 2. AN EIGHT-STATE, ELEVEN-NEIGHBOR, REVERSIBLE CELLULAR AUTOMATON

It is easier to describe our CA as a stylized version of the BBM. Then, it will be clear how to define the formal local transition map of this CA.

First, we consider a "universe" in which there are only two kinds of particles: (1) moving particles, which never stop; we call them "balls"; and (2) infinitely massive particles, which never move, even when kicked by moving particles; we call them "mirrors" (we can also consider the last particles as "force fields".) There are only two possible directions for moving balls, which are perpendicular to each other. We indicate them by right-to-left and down-to-up vectors. All balls have constant and equal velocity. The "universe" is a Euclidean two-dimensional plane. In this plane, there are Cartesian coordinate systems in which at some moments all particles are at points with integer coordinates. If we consider one of these moments as the origin of time, then at times, $\ldots, 0, 1, 2, \ldots$ all particles have integer coordinates, and these numbers completely determine the system; so we can consider our "universe" as a system in which both space and time are discrete. The possible places for particles constitute nodes of a grid in the plane, and at each moment each particle is at some node of this grid. One of the specific features of this system is that at a node there may be particles of different kinds simultaneously (two balls are of different kinds if their directions are different).

So, in our CA eight different states can occur at a node. These states are represented in Figure 1. In the figures, we represent each node by a square with this node as its center.

The state 0 represents the absence of any particle (in cellular automata terminology, it is the quiescent state). The states 1 and 2 represent balls in different directions. The state 4 represents a mirror. The states 3 and 5–7 represent the four possible occurrences of more than one particle at a node.

Now we explain the kinematic laws in this "universe." At each time step, the balls move one node, according to their directions. Figure 2 represents various possible cases. This law is violated in a few cases which we will describe later.

Mirrors never move, but they change the "local geometry" of their region, that is, the balls move according to different laws near the mirrors. When one or two balls kick a mirror (this will be denoted by the states 5, 6, or 7), then their directions change. The first three rows of Figure 3 represent this situation, and other rows represent possible cases for kicking. The law defined by last three rows would be violated in a few cases that we will describe later.

If there is more than one adjacent mirror in a region, then new special laws govern there. Two or more adjacent mirrors will shift the path of a ball. This law is expressed in Figures 4, 5, and 7. In these and other figures a dot in a cell indicates the absence of mirror at that node, that is, that cell

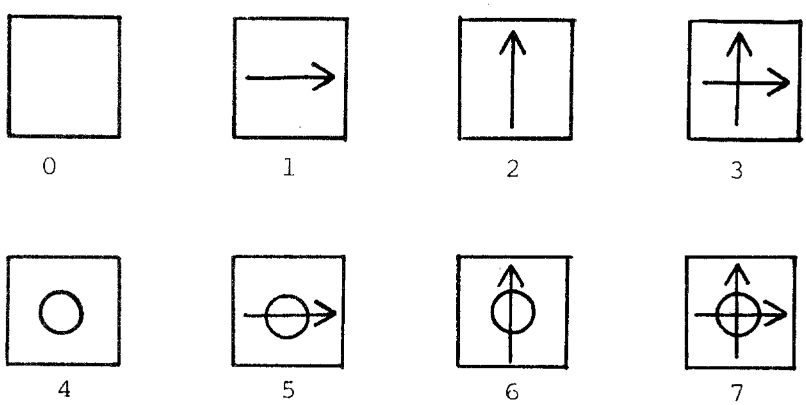


Fig. 1.

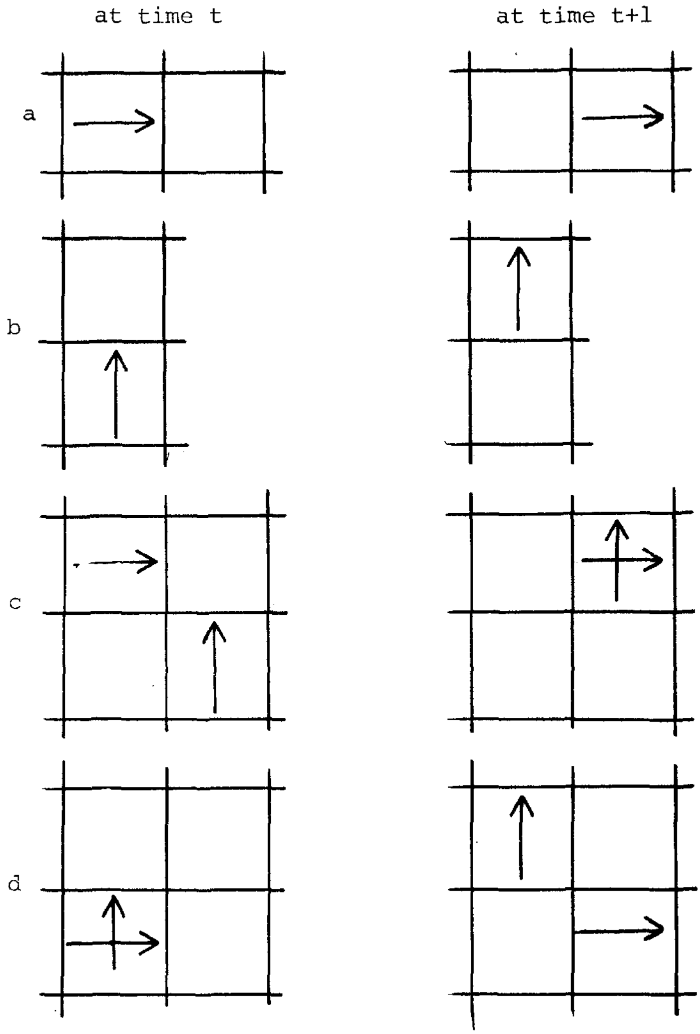


Fig. 2.

is in one of the states 0–3; in these figures, the state of unspecified cells may be any state.

There is a superposition rule, i.e., for determining the state of the "universe" at a moment, all the above laws apply simultaneously. It can easily be checked that these laws are mutually consistent with each other. For example, from the laws indicated by Figures 4c and 4d, one has the law indicated by Figure 8.

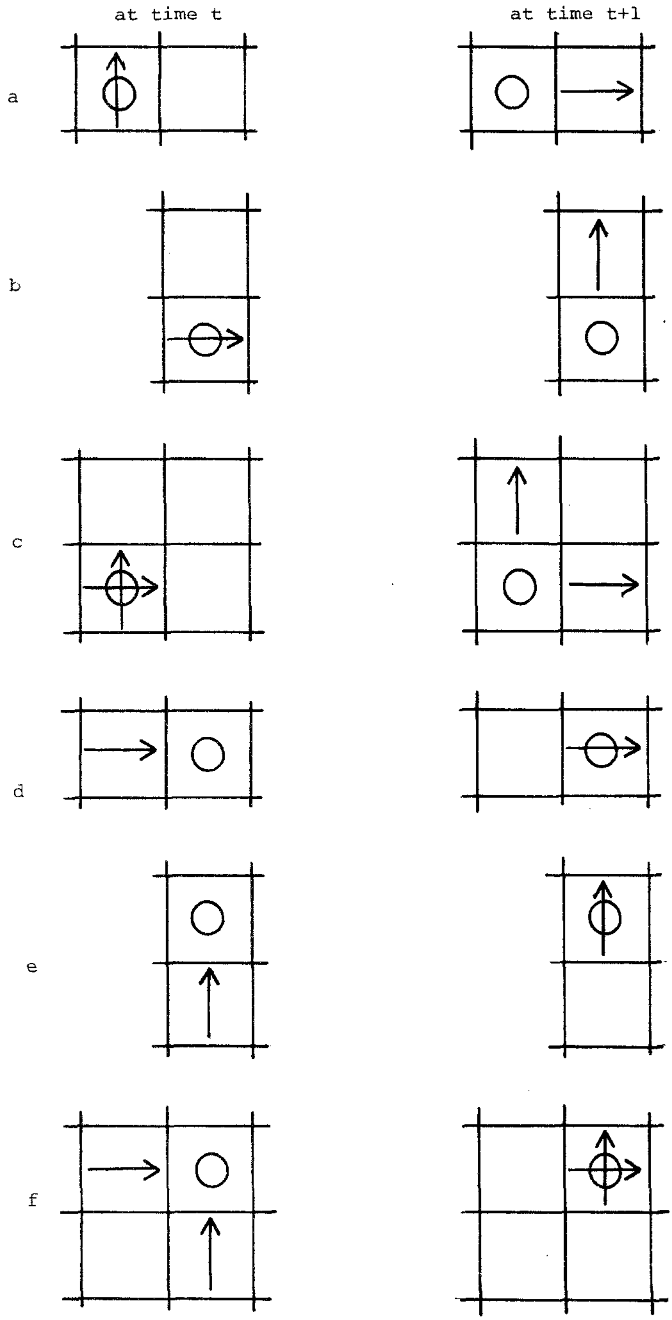


Fig. 3.

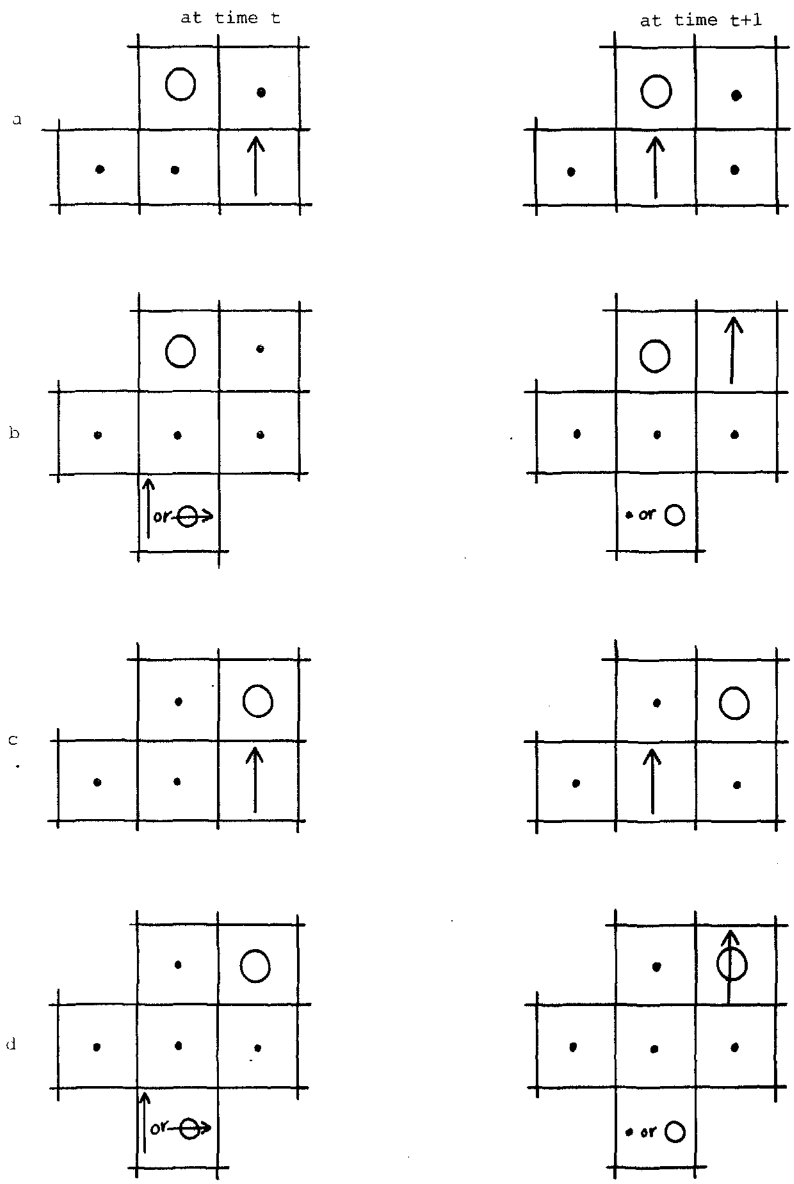


Fig. 4.

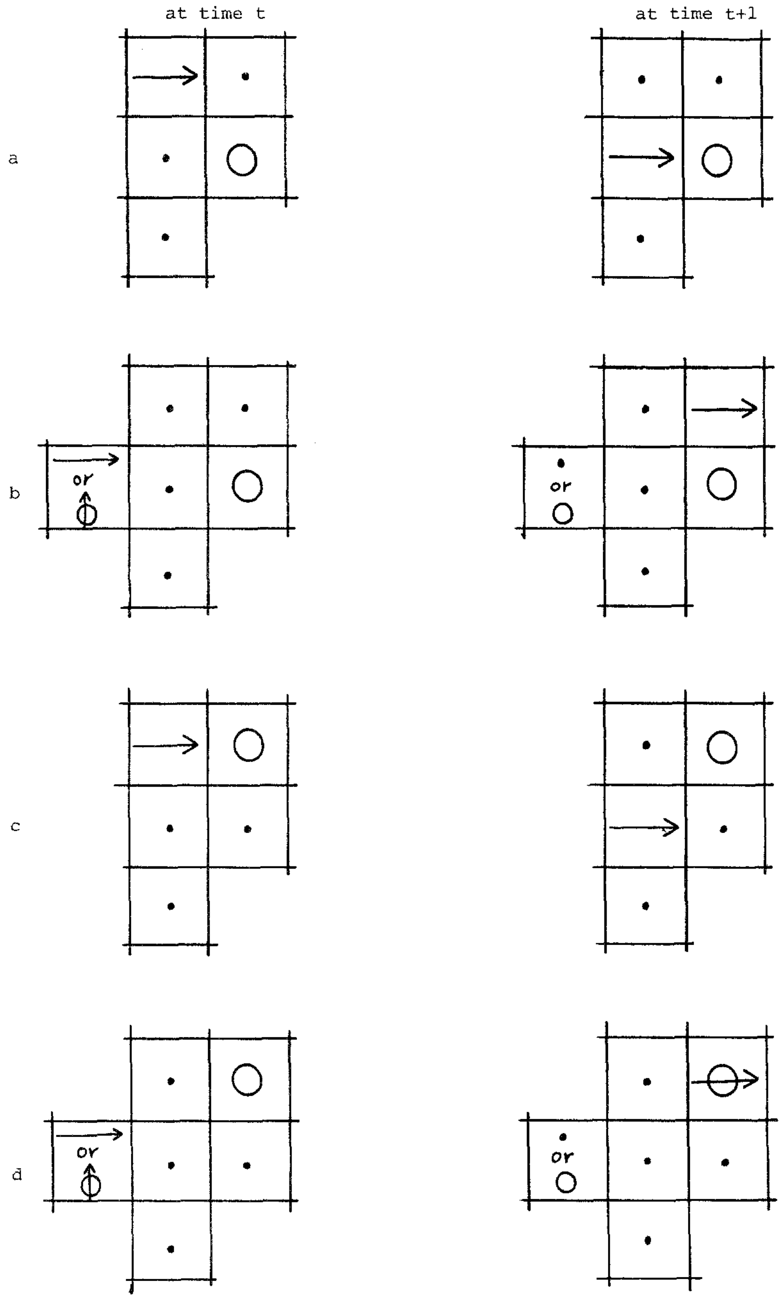


Fig. 5.

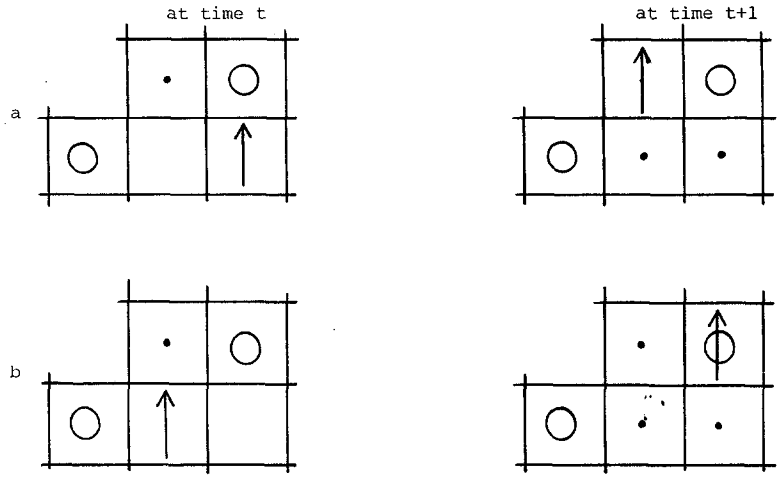


Fig. 6.

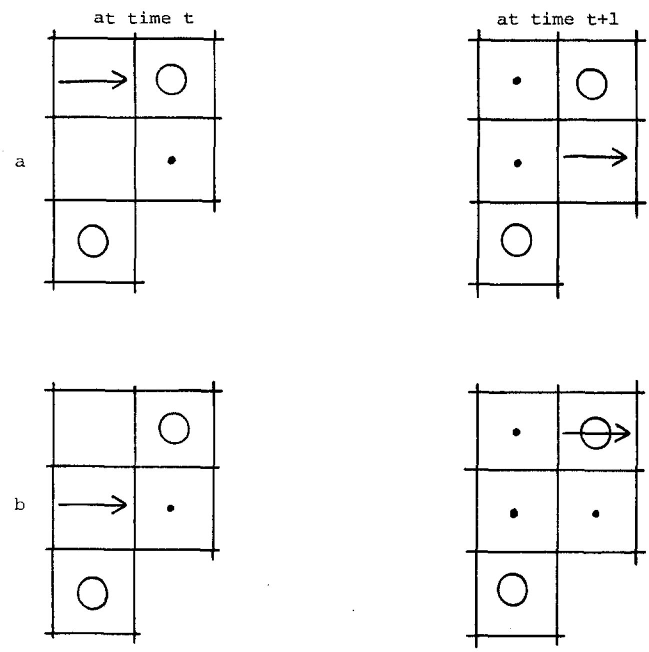


Fig. 7.

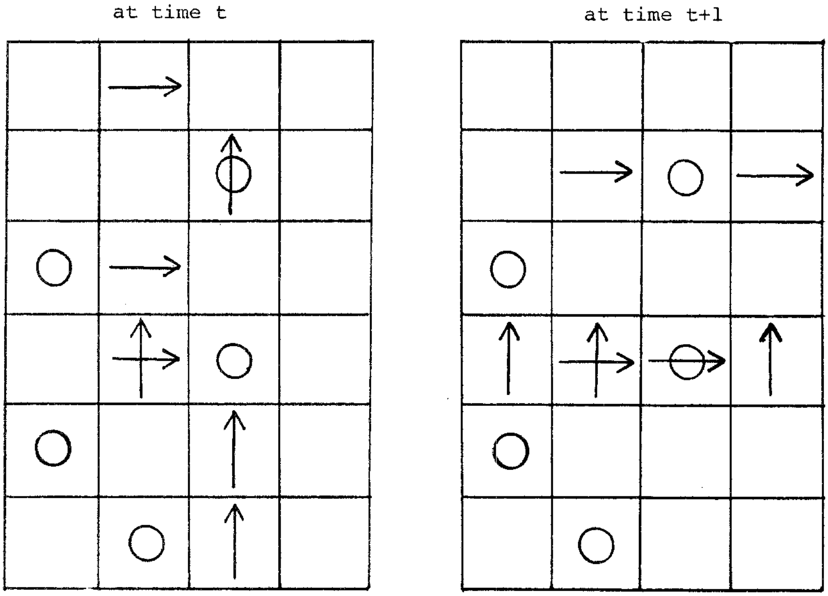


Fig. 8.

Now, it can be easily verified that the state of a cell at time $t+1$ only depends on itself and the states of all ten of its adjacent cells, which indicated in Figure 9, at time $t$. Since there are eight possible states for each cell, the kinematic laws of our "universe" can be expressed as local transition maps of an eight-state, 11-neighbor cellular automaton.

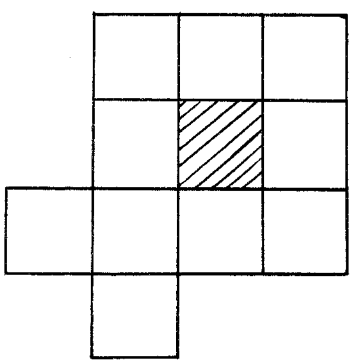

Fig. 9.

## 3. UNIVERSALITY

In this section we prove that our system is computationally universal. This means that it can simulate all computational processes, i.e., it can simulate a universal digital computer. For this, it is enough to show that this system can simulate the BBM, because it has been proved that this model is computationally universal (Fredkin and Toffoli, 1982).

The BBM has been described in Fredkin and Toffoli (1982) and Margolus (1984), and we refer the reader to these references for its definition. States 1 and 2 represent balls moving left-to-right and down-to-up, respectively. The state 4 clearly acts as a reflector in the BBM. Figures 6 and 7 represent the collision of two balls. We must also show that in our system balls can move from right to left and from up to down. Figures 4a and 4c show that a horizontal string of every other mirror can transform a ball from right to left; similarly, Figures 5a and 5c show that a vertical string of every other mirror can transform a ball from up to down. If another ball is to cross over these strings, then according to Figures 4b, 4d, 5b, and 5d it will be shifted to another row or column, but with mirrors suitably placed it can go back to its original row or column. Thus, as far as its computational universality is concerned, the BBM can be simulated by our system.

Here we explicitly represent two important gates in reversible computing. Figure 10 represents a delay gate and its action. Figure 11 represents a switch gate and its action. (Note the use of the superposition rule in Figures 10 and 11.)

## 4. REVERSIBILITY

There are many definitions of "reversibility" in cellular automata. These definitions and their relations to each other have been discussed in Di Gregorio and Trautter (1975). Here we choose the most restrictive definition. A CA is *reversible* if every configuration of it has a unique predecessor. More formally, let $\mathscr{C}$ be the set of all configurations of the CA. Let $F$ be the parallel map of this CA. Now, if $F$ is a bijective (i.e., one-to-one and onto) map from $\mathscr{C}$ to $\mathscr{C}$, then it is said that this CA is reversible. A configuration is said to be finite if it has only a finite number of nonquiescent cells. In our automaton every finite configuration has a unique predecessor that is also a finite configuration. This is obvious because of the fact that our system has a "conservation law": the total number of balls and mirrors is fixed.

Let $\tau$ be the local transition map of our CA defined by Figures 2–7 and the superposition rule. Let $\sigma$ be the local transition map defined by interchanging configurations at time $t$ and $t+1$ in Figures 2–7 and the

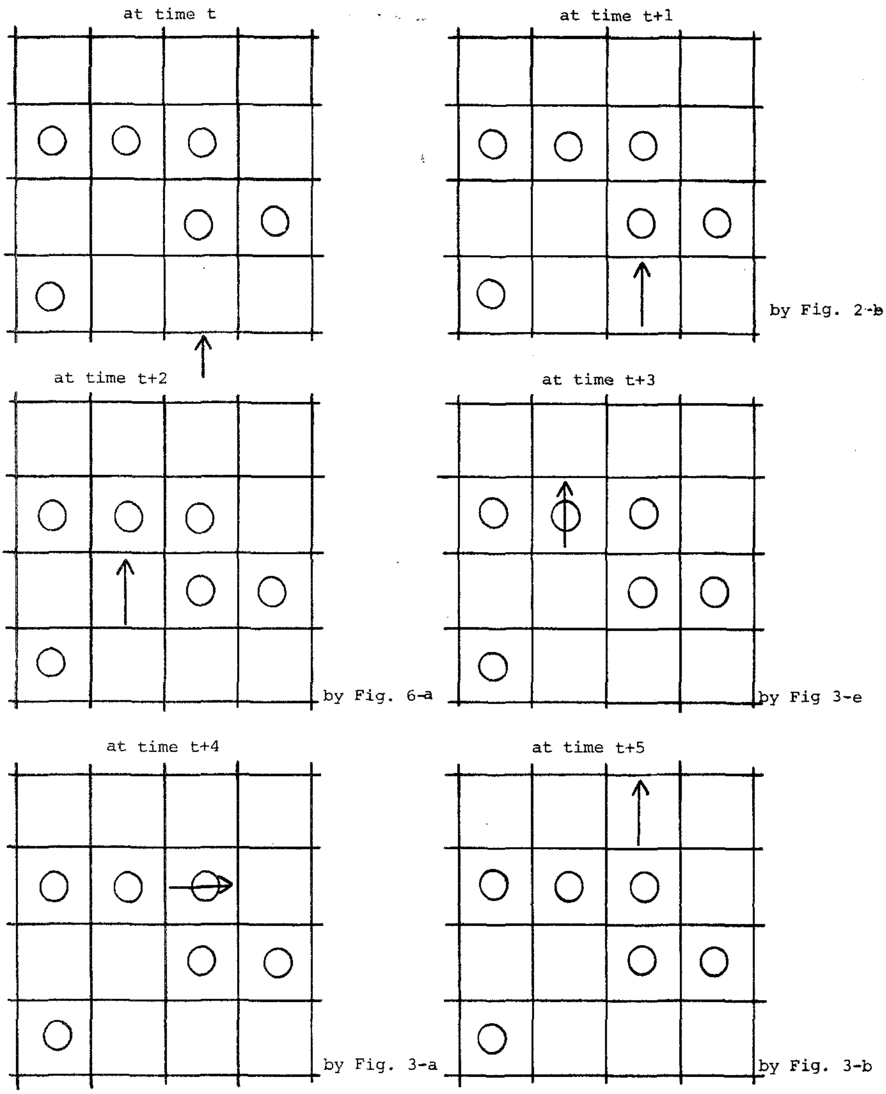


**Fig. 10.**

superposition rule. The transition map $\sigma$ is well defined, because no overlapping can occur.

If we represent the parallel maps associated with $\tau$ and $\sigma$ by $F$ and $G$, respectively, then it is easily verified that for every configuration $C$ (finite or infinite), we have $F(G(C)) = C$ and $G(F(C)) = C$. Thus, $F$ is a bijective map and our CA is reversible.

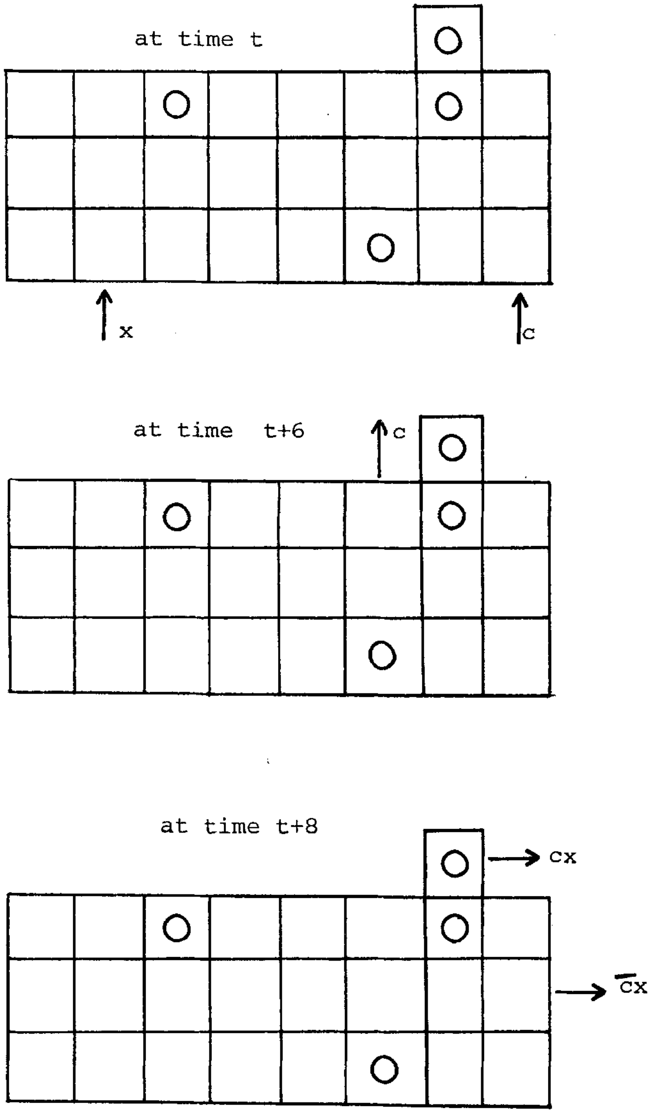


Fig. 11.

## 5. AN INTRACTABLE PROBLEM

The complexity of a solvable problem by a computational procedure can be measured by the number of steps or the volume of storage capacity needed for performing that procedure. These quantities are said to be the time and space complexity of that problem, respectively. These measures are expressed as functions of the size $n$ of instances of that problem; i.e., the amount of information used in specifying that instance. One may then define several classes of problems. The first, denoted by P, are those that can be solved in a time polynomial in $n$. The second, denoted by PSPACE, are those that can be solved in a space polynomial in $n$.

It is widely accepted that only those problems have efficient algorithms that are in P. It is clear that P is a subclass of PSPACE, but there is plenty of evidence that P$\neq$PSPACE. There are plausible examples of problems that are in PSPACE, but they do not seem to belong to P; these problems are called PSPACE-complete, because every instance of a problem in PSPACE can be decoded efficiently (i.e., in a time polynomial in the size of that instance) to an instance of these problems. So, every procedure for solving a PSPACE-complete problem actually mimics the operation of a universal computer with polynomial-bounded storage capacity. If, as is widely conjectured, it is true that P$\neq$PSPACE, then all PSPACE-complete problems are actually intractable. [For a comprehensive study of these notions see Hopcroft and Ullman (1979).]

We will show that the following problem, which occurs naturally in our automaton, is PSPACE-complete, and consequently, is intractable: there is a specific distribution of balls (in various directions); can we place some mirrors in the grid so that these balls will eventually reach a predetermined distribution? Before proving this, we give some necessary definitions.

In the sequel, all configurations are finite. By considering a coordinate system for each cell, we can give its coordinates as its address. A *window* is a finite collection of addresses of cells; and we say that a particle is in the *scope* of a window if the address of its cell is in that window. If $C$ is a configuration and $W$ is a window, the *restriction* of $C$ to $W$, denoted by $C|W$, is the configuration obtained from $C$ by eliminating all particles that are not in the scope of $W$; so, in $C|W$ the state of all cells whose addresses are not in $W$ is quiescent. If $C$ and $D$ are two configurations, then $C+D$ is the configuration obtained by superposing $C$ and $D$; this is only defined when no cells of $C$ and $D$ have the same particles.

Let $C$ and $D$ be two (finite or infinite) configurations. We say that $D$ is *accessible* from $C$, and write $C \vdash D$, if there is a positive integer $k$ such that $F^k(C)=D$, where $F$ is the parallel map of the automaton and $F^k(C)$ indicates $k$ times application of $F$ to $C$. In this case we also say that $D$ is accessible from $C$ after $k$ steps.

A *mirror replacement* (m.r.) is a finite configuration in which there are no balls; and a *ball replacement* (b.r.) is a finite configuration in which there are no mirrors.

Every configuration $C$ can be considered as a function from $Z \times Z$ to the set of states, where $Z$ denotes the set of integers. Each element $c=(i,j)$ from $Z \times Z$ represents the address of a cell and $C(c)$ is the state of that cell. The support of a configuration $C$ is the set of all $c$'s such that $C(c) \neq 0$; and the scope of finite configuration $C$ is the smallest rectangular set of adjacent cells that contains the support of $C$.

*Mirror Replacement Problem* (MRP)

*Instance.* Two finite sets

$$\{A_1, \ldots, A_n\} \quad \text{and} \quad \{B_1, \ldots, B_n\}$$

of b.r.'s and a set $\{W_1, \ldots, W_n\}$ of windows.

*Question.* Is there an m.r. $M$ such that for each $i$, $1 \leq i \leq n$, we have $A_i + (M|W_i) \vdash B_i + (M|W_i)$?

*Lemma.* For every m.r. $M$ with scope $S$ there are two sets of b.r.'s $\{A_1, \ldots, A_k\}$ and $\{B_1, \ldots, B_k\}$ and a set of windows

$$\{W_1, \ldots, W_k\}$$

that completely determine $M$; i.e., $M$ is the only m.r. with scope $S$ such that for every $i$, $1 \leq i \leq k$, we have

$$A_i + (M|W_i) \vdash B_i + (M|W_i)$$

Furthermore, $k$ and the sizes of these b.r.'s and m.r.'s are linear in the size of $M$.

*Proof.* We arrange the cells in $S$ as a sequence

$$C_1, C_2, \ldots, C_k$$

For every cell $c_i$, let $e_i$ be its eastern neighbor and $s_i$ its southern neighbor, as indicated in Figure 12. If $c_i$ contains a mirror, then $W_i$ consists

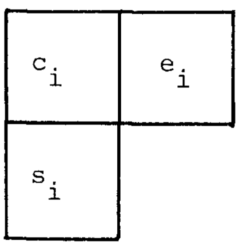


**Fig. 12.**

of the addresses of $c_i$ and $e_i$, and the b.r.'s $A_i$ and $B_i$ are defined for arbitrary cell $c$ as follows:

$$A_i(c) = \begin{cases} 2, & c = c_i \\ 0, & c \neq c_i \end{cases}$$
$$B_i(c) = \begin{cases} 1, & c = e_i \\ 0, & c \neq e_i \end{cases} \tag{1}$$

If $c_i$ is in the quiescent state, then

$$A_i(c) = \begin{cases} 1, & c = c_i \\ 0, & c \neq c_i \end{cases}$$

and if $e_i$ is also in the quiescent state, then $W_i$ consists of addresses of $c_i$ and $e_i$, and $B_i$ is defined by (1); if $e_i$ contains a mirror and $s_i$ is in the quiescent state, then $W_i$ consists of addresses of $c_i$, $e_i$, and $s_i$ and

$$B_i(c) = \begin{cases} 1, & c = s_i \\ 0, & c \neq s_i \end{cases}$$

and finally, if both $e_i$ and $s_i$ contain a mirror, then $W_i$ consists of addresses of $c_i$ and $c_i$, and $B_i$ is defined by (1).

Now, it is easily verified that the configurations $A_i$, $B_i$ and $W_i$ satisfy the requirements of the lemma. ■

*Theorem 1.* The MRP is PSPACE-complete.

*Proof.* Let $T$ be a $P(n)$-space bounded, deterministic Turing machine, where $P(n)$ is a polynomial (for a comprehensive study of this and subsequent notions see Hopcroft and Ullman, 1979). Since our cellular automaton is computationally universal, the behavior of $T$ can be simulated by this CA. We can assume that there is a standard procedure for encoding inputs and accepting and rejecting terminations of $T$ as special finite configurations in this CA. So, for every input $w$ of $T$, there are b.r.'s $I$ and $A$ that encode input $w$ and accept termination of $T$ over $w$, respectively. The sizes of $I$ and $A$ are both bounded by polynomial functions of $n$, where $n$ is the length of $w$. There is also an m.r. $M$ that represents action of $T$ and supplies adequate space for computation of $T$ over $w$; the size of $M$ is linear in $P^2(n)$. Let $S$ be the scope of $M$, and $W$ the window consisting of addresses of cells of $S$. [A similar construction is used in Berlekamp *et al.* (1982) and Nourai and Kashef (1975).]

Now, by previous lemma, there are sets $\{A_1, \ldots, A_k\}$ and $\{B_1, \ldots, B_k\}$ of b.r.'s and a set

$$\{W_1, \ldots, W_k\}$$

of windows that determine $M$ uniquely.

The sizes of these sets are polynomial functions of $n$. Now, $T$ accepts $w$ iff the MRP associated with sets $\{I, A_1, \ldots, A_k\}$ and $\{A, B_1, \ldots, B_k\}$ of b.r.'s and $\{W_1, \ldots, W_k\}$ of windows is solvable. So, the MRP is PSPACE-hard.

It is obvious that the MRP itself is in PSPACE. For solving any instance of this problem, it suffices to consider all possible m.r.'s whose scope is determined by the given window, separately. ∎

## 6. DECIDABLE AND UNDECIDABLE PROBLEMS

By an undecidable problem we mean a problem for which there is no computational procedure (algorithm) that can solve every instance of that problem. There are many well-known undecidable problems. One of them is the "dying problem" in universal cellular automata; i.e., given a finite configuration, does it eventually lead to the quiescent configuration? For example, it has been proved that the dying problem for the "game of Life" is undecidable; the "game of Life" is a two-state, nine-neighbor, two-dimensional, irreversible, universal cellular automaton (Berlekamp *et al.*, 1982).

In our reversible cellular automaton the dying problem is clearly decidable, because no nonquiescent configuration can lead to the quiescent configuration.

The dying problem is a special case of a more general problem:

*Accessibility Problem* (AP)

*Instance.* Two configurations $C$ and $D$ of a cellular automaton.

*Question.* In this automaton, is it true that $C \vdash D$?

It is clear that for all cellular automata with the dying problem undecidable, the AP is undecidable, too. But for our reversible cellular automaton this problem is decidable, because, in this CA, the number of particles is invariant, and for every finite configuration $C$, the evolution of $C$ by our CA leads either to a periodic sequence of configurations or to configurations whose scopes are arbitrarily large. The latter occurs only when some balls of $C$ go beyond the scope of mirrors of $C$. One needs a finite time to decide whether $D$ is accessible from $C$ or not.

But why is the AP for this reversible cellular automaton decidable, while for some irreversible cellular automata it is undecidable? In all undecidable problems some instances of the problems need an unbounded amount of memory capacity; and in our CA the accessible memory is actually a part of the initial configuration, and because of the "conservation law" of this automaton, this memory capacity cannot be increased during the evolution of the automaton, so it is bounded. The situation is similar to real computing machines, where the amount of memory capacity is restricted by physical constraints. This is further evidence that the reversible computing reveals the physical aspects of computing.

Now that the AP of our automaton is decidable, what about its complexity?

*Theorem 2.* The AP of our reversible universal cellular automaton is PSPACE-complete.

*Proof.* For every polynomial $P(n)$, every $P(n)$-space bounded Turing machine $T$, and every input of $w$ of length $n$ for $T$, there are finite configurations $C$ and $D$ such that $C$ represents the input $w$ and the transition rules of $T$ and also provides permissible memory capacity, and $D$ represents accepting termination of $T$ on $w$. Now, $T$ accepts $w$ iff $D$ is accessible from $C$.

Furthermore, it is obvious that the AP is in PSPACE. Therefore, the AP is PSPACE-complete. ■

Now we consider a similar problem, which is a special case of the MRP. There are two b.r.'s $B$ and $B'$; is there an m.r. $M$ such that $B+M \vdash B'+M$? (note that there are no windows). It can be shown that this problem is always decidable; if the numbers of balls of $B$ and $B'$ are equal, then the answer is yes; otherwise, no. If we add a window $W$ and want $B+M|W \vdash B'+M|W$, then the new problem is clearly in PSPACE, and it seems that it is actually a PSPACE-complete problem, but we have no proof.

## ACKNOWLEDGMENT

This research was supported by the Atomic Energy Organization of Iran.

## REFERENCES

Bennett, C. H. (1973). Logical irreversibility of computation, *IBM Journal of Research and Development*, **17**, 525–532.

Berlekamp, E. R., Conway, J. H., and Guy, R. K. (1982). *Winning Ways*, Chapter 25, Academic Press, New York, 1982.

Di Gregorio, S., and Trautter, G. (1975). On reversibility in cellular automata, *Journal of Computer and System Sciences*, **11**, 382–391.

Fredkin, E., Toffoli, T. (1982). Conservative logic, *International Journal of Theoretical Physics*, **21**, 219–253.

Hopcroft, J. E., and Ullman, J. D. (1979). *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, Massachusetts, 1979.

Landauer, L. D. (1961). Irreversibility and heat generation in the computing processes, *IBM Journal of Research and Development*, **5**, 183–191.

Margolus, N. (1984). Physics-like models of computing, *Physica*, **10D**, 81–95.

Nourai, F., and Kashef, R. S. (1975). A universal four-state cellular computer, *IEEE Transactions on Computers*, **C-24**, 766–776.

Richardson, D. (1972). Tesselations with local transformation, *Journal of Computer and System Sciences*, **6**, 373–388.

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

Toffoli, T. (1980). Reversible Computing, Technical Memorandum MIT/LCS/TM-151, MIT Laboratory for Computer Science.