Universal computation in few-body automata

Michael Biafore*

Department of Physics, Massachusetts Institute of Technology,

Cambridge, MA 02139, USA

Abstract

Few-body automata are a class of cellular automata. They were specifically developed to investigate the possibility of implementing a new generation of cellular automata machines using dense arrays of nanometer-scale device-cells. In this paper we try to determine how many states per cell few-body automata need in order to perform universal computation. Simulation theorems describing the space, time, and state-set complexity of simulating d-dimensional conventional cellular automata (CA) with N-body automata are proven. It is shown that for $d = 1$ there exists a computation-universal 2-body automaton requiring 5.81 bits of state per cell; for $d = 2$, it is shown that there exists a computation-universal 2-body automaton requiring 2 bits of state per cell. These results suggest that the number of bits per cell required to attain computation-universality will not be an obstacle to CA-like computation at nanometer scales.

1 Introduction

Responding to perceived limits to the continued downscaling of microprocessor architectures [5, 11, 25], a number of device physicists [4, 31, 15, 30] have proposed an alternative architecture based on cellular automata. Specifically, they envision dense arrays of nanometer-scale devices communicating with one another via *direct physical interactions* rather than through wired

*Internet address: `biafore@im.lcs.mit.edu`

interconnections. The devices might consist of semiconductor heterostructures, molecular switches, or some other nanometer-scale structure capable of changing state in response to interactions with neighboring structures. For certain interactions among cells, the array might be made to function as a cellular automaton in which each device acts as a single cell.

Various researchers have proposed that devices be coupled via resonant quantum tunneling [4, 31, 13], Coulomb interaction effects [21, 30], or soliton switching in π-conjugated polyenes [9]. If such proposals are ever realized, the resulting cellular automata machines would be unlike any existing cellular automata machine[34, 35, 36]. The dynamics of the cellular automaton would no longer be determined by an arbitrarily definable lookup table; instead, it would be completely determined by the particular physical interactions [29] between cells.

Few-body automata [7] are a class of cellular automata developed to investigate the computational feasibility of such proposals.

The geometric form of few-body automata is defined to reflect certain unavoidable constraints imposed by the finite range of the physical interactions between cells. Their computational structure places them in the class of lattice-gas [18, 17, 14] (or partitioning [27]) cellular automata. In an N-body automaton, each neighborhood consists of a cluster of N cells—two cells are neighbors if and only if they are in the same cluster, just as the neighborhood of a hexagonal lattice-gas consists of the $N = 6$ sites incident on a given node of the hexagonal lattice.

In this paper, we are principally interested in putting an upper bound on the number of states required to attain computation universality in the limiting case $N = 2$. The importance of this limit has two sources. First, the physics of two interacting devices is both experimentally the most accessible and theoretically the most tractable. Secondly, from a computational perspective the case of $N = 2$, where each cell has only a single neighbor aside from itself, is the most difficult limit in which to achieve computation universality. Therefore, the upper bounds we will obtain on the number of states required for $N = 2$ automatically represent upper bounds for all N-body automata with $N \geq 2$.

In addition, these bounds can be used to evaluate the practicality of using specific nanometer-scale devices as cells of a CA. Some of the proposed coupling mechanisms (such as single-electron tunneling[21]) are limited by the underlying physics to providing only a single bit per cell [7, 21, 30]; none of them can provide more than a few bits per cell. Consequently, although one can easily show that even in the limit of 2-body cellular automata universal computation is possible, the question of practical relevance is, "How many bits per cell will be required"?

To address this question, in section 3 we establish a general theorem describing the efficiency with which a 2^d-body automaton in d-dimensions can simulate an arbitrary conventional cellular automaton (CA). By applying this simulation theorem to conventional CA known to be computation-universal, we establish the existence of a universal 2-body automaton with 5.81 bits per cell. Finally, we show how 4-body automata can be systematically simulated by 2-body automata. Using this technique, we obtain a universal 2-body automaton requiring just 2 bits per cell.

2 N-body cellular automata

To begin, we define explicitly what we will mean by "conventional" (or "von Neumann") cellular automata, and explain why they are not a satisfactory tool for analyzing the CA-like properties of arrays of interacting devices. Conventional CA [37] reside on a d-dimensional cartesian lattice. The neighborhood of a given cell is defined to consist of the $(2r+1)^d$ cells within some radius r. (For example, in $d = 2$, the case $r = 1$ gives the well-known Moore neighborhood [35].) The next state of the cell then depends on an evolution operator U with domain $Q^{(2r+1)^d}$, where Q is the finite set of states each cell may occupy.

Definition 2.1 *On a d-dimensional rectangular lattice, a* conventional cellular automaton *with radius r is a triple $\langle U, Q, r\rangle$, where Q is the finite set of states that each cell can occupy, r is the radius of the neighborhood and $U : Q^{(2r+1)^d} \to Q$ is the local evolution operator.*

That is, at time $t+1$, the state $a^{t+1}_{i_1,i_2,\ldots,i_d}$ of the cell with cartesian coordinates $(i_1, i_2, \ldots, i_d)$ is related to its own state and those of neighboring cells by the relation

$$a^{t+1}_{i_1,i_2,\ldots,i_d} = U \begin{pmatrix} a^t_{i_1,i_2,i_3\ldots,i_d} \\ a^t_{i_1+1,i_2,i_3\ldots,i_d} \\ a^t_{i_1-1,i_2,i_3\ldots,i_d} \\ \vdots \\ a^t_{i_1+r,i_2,i_3\ldots,i_d} \\ a^t_{i_1-r,i_2,i_3\ldots,i_d} \\ \vdots \\ a^t_{i_1+r,i_2+r,i_3+r\ldots,i_{d+r}} \\ a^t_{i_1-r,i_2-r,i_3-r\ldots,i_{d+r}} \end{pmatrix}. \tag{2.1}$$

The spacetime diagram for a one-dimensional conventional cellular automaton with $r = 1$ is show in figure 1.

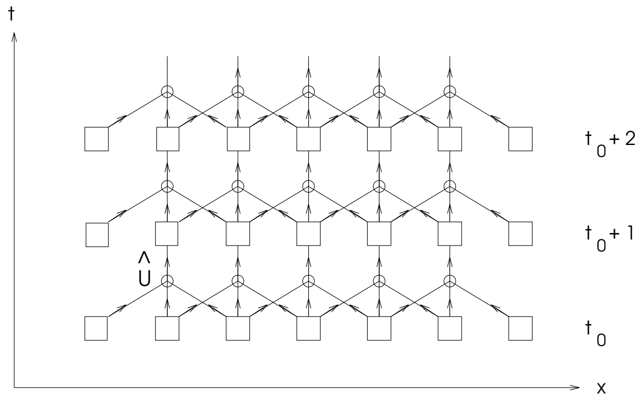

Figure 1: Time evolution of a conventional CA in one dimension.

Unfortunately, for the application device physicists are proposing, where cells are coupled by real physical interactions (such as Coulomb dipole interactions [30, 7]), conventional CA are not a suitable model. Note that cellular automata are characterized by three forms of discreteness:

1. Each cell can take on only a finite number of states,

2. The state of each cell changes at discrete time intervals and,

3. The state of each cell at the following time-step is completely determined by a well-defined (usually small) subset of nearby cells, its "neighbors".

If we attempt to construct a conventional CAby simply arrange nanometer-scale devices in a rectangular array, then it becomes difficult if not impossible to retain properties 2 and 3. Property 2 is difficult to satisfy because the cells are coupled by physical interactions, with the result that state transitions occur continuously rather than at controlled discrete time-intervals. Property 3 is similarly difficult to satisfy. Although the effect of distant cells decreases with separation (e.g., the Coulomb dipole potential [24] falls off as $\sim 1/r^2$), in most cases the next state of each cell will depend on a large, rapidly fluctuating set of cells. The resulting dynamics may be of interest in its own right, but other than its spatial discreteness, it has little in common with cellular automata. Until we find a format to constrain the dynamics of such device arrays so that they have all three forms of discreteness essential to cellular automata, we will not be able to analyze their computational properties within the framework of cellular automata theory.

For certain kinds of interactions between cells, we can impose the required constraints by imposing the structure of N-*body* automata.

1. Since we want to use physical interactions between cells to implement the cellular automaton, instead of adopting the definition of conventional cellular automata, we let the form of the interactions determine the form of the cellular automaton.

2. Figure 2 is a schematic representation of the most general possible physical interaction between six entities. Fortunately, the interactions between devices are often more tractable than the most general case.

3. If an interaction is not too singular when the interacting particles approach one another, and if it dies off fast enough as they are separated to infinity, then one can show rigorously that it is valid to view the complex interaction process in figure 2 as the composition of a number of independent N-body interactions, as in figure 3. Such interactions are said to be *regular*[32]. For example, L. D. Fadeev[12] and Hepp[20] have shown that a scalar 2-body interaction is regular if it satisfies

$$\lim_{r \to 0} r^{\frac{3}{2}-\epsilon} V(r) = 0, \tag{2.2}$$

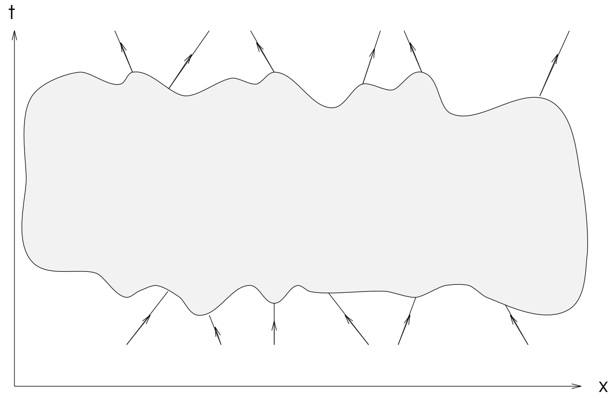

Figure 2: Generic process with six input states and six output states.

and

$$\lim_{r\to\infty} r^{3+\epsilon} V(r) = 0. \tag{2.3}$$

Regular interactions are much more amenable to theoretical analysis because the diagrammatic and algebraic tools of many-body theory can then be applied.

However, the spacetime structure in 3 are still too irregular in space, time, and number of interacting entities to be regarded as any kind of cellular automaton. But if we begin with a spacetime structure like that in figure 3 and discretize the times and places at which interactions take place, we obtain the one-dimensional, 2-body automaton shown in figure 4. This should be contrasted with the corresponding spacetime diagram of for a one-dimensional conventional CAin figure 1.

Note that in a 2-body automaton, cells interact in pairs—the next state of each cell depends on its own current state and, on alternate time steps, the state of either its left or right neighbor. All 2-body automata have this spacetime structure, only the particular rule $\hat{U}$: $(a_i^t, a_{i+1}^t) \to (a^{t+1}, a_{i+1}^t)$ mapping the states of the two cells at time t into those at $t+1$ varies.

Now that we have a qualitative idea of what few-body automata are, we can give a definition that is formal enough to let us prove the sim-

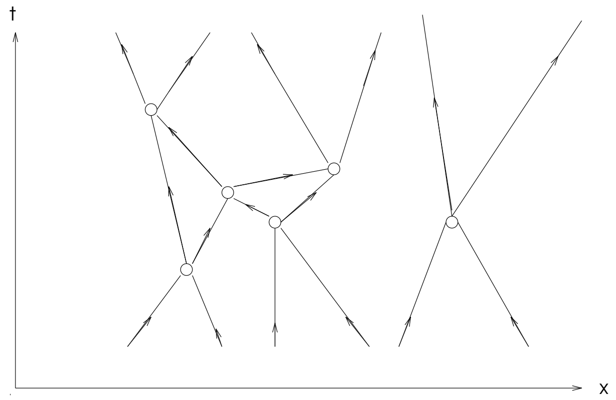

Figure 3: Description of a regular physical interaction process by an N-body diagram.

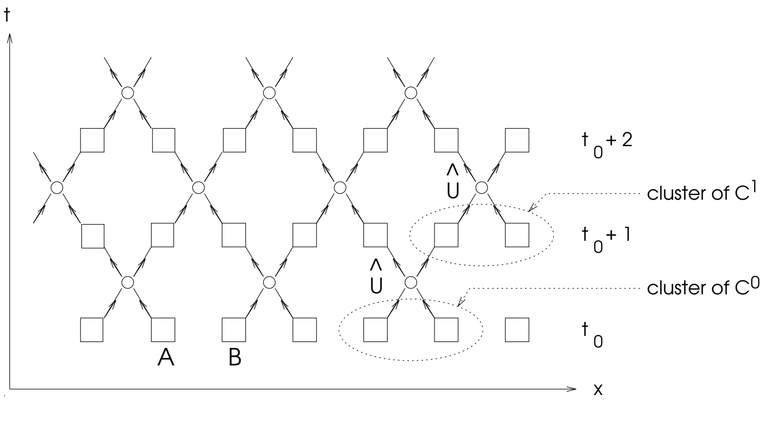

Figure 4: Time evolution of a 2-body CA in one dimension is identical to that of a one-dimensional lattice gas.

ulation theorem. Quite general definitions of few-body automata are possible since one can vary many parameters: the number N of cells clustered together at each step, the shape of the clusters, and the temporal periodicity of the spacetime pattern. We restrict ourselves to the most straightforward definition.

In d dimensions, we consider only clusters which are hypercubes with two cells per side, so that $N = 2^d$. Furthermore, as in figure 4, we consider only the case where the spatial periodicity is two—that is, after two time steps, the cells are again "clustered" in the same way. In figure 4, the clusters are the pairs of cells entering and emerging from the evolution operator $\hat{U}$.

On the even time steps, we will denote the collection of all clusters by $\mathcal{C}^0$, and on the odd steps by $\mathcal{C}^1$. The clusters of $\mathcal{C}^0$ and $\mathcal{C}^1$ consist of d-dimensional cubes with two cells along each edge

$$\mathcal{C}^0 = \{\mathcal{C}_{2i_1,2i_2,\ldots 2i_d} | i_k \in Z\}, \tag{2.4}$$

and

$$\mathcal{C}^1 = \{\mathcal{C}_{2i_1+1,2i_2+1,\ldots 2i_d+1} | i_k \in Z\}, \tag{2.5}$$

where

$$\mathcal{C}_{i_1,i_2,\ldots i_d} = \{i_1 + \tau_1, i_2 + \tau_2, \ldots, i_d + \tau_d | i_k \in Z \,, \tau \in \{0,1\}\} \tag{2.6}$$

denotes the d-dimensional cube of cells whose corner closest to the origin lies at rectangular coordinate $(i_1, i_2, \ldots, i_d)$. It is important to note that the clusters of $\mathcal{C}^1$ are offset by $(1, 1, \ldots, 1)$ relative to the clusters of $\mathcal{C}^0$.

The 2^d-body cellular automata we will use to simulate conventional automata can be defined in terms of the clusterings $\mathcal{C}^0$ and $\mathcal{C}^1$.

Definition 2.2 *A d-dimensional* 2^d-body *cellular automaton is a pair* $\langle \widehat{U}, \widehat{Q} \rangle$, *where* $\widehat{Q}$ *is the finite set of states that each cell can occupy and the automorphism* $\widehat{U} : \widehat{Q}^{2^d} \to \widehat{Q}^{2^d}$ *is a local evolution operator which updates each cluster of* 2^d *cells belonging to the clusterings* $\mathcal{C}^0$ *and* $\mathcal{C}^1$.

In $d = 1$, this definition simply reproduces the 2-body automaton shown in figure 4; in $d = 2$, it defines the class of 4-body automata shown in figure 5.

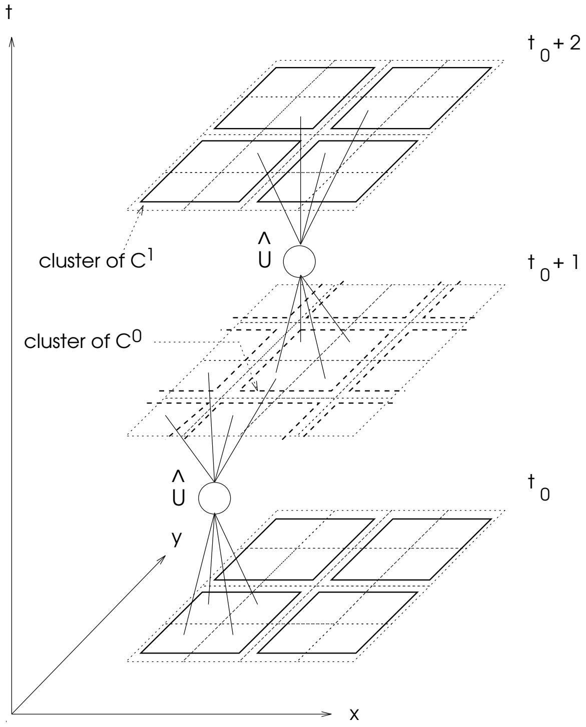

Figure 5: Four-body cellular automaton with 2×2 cluster neighborhoods. Clusters of $\mathcal{C}^0$ are shown as dashed squares; clusters of $\mathcal{C}^1$ are solid; the cells are shown as dotted boxes.

2.1 Simulation by Contraction Automata

We will show how any d-dimensional conventional cellular automaton $\langle U, Q, r\rangle$ can be simulated by a 2^d-body automaton $\langle \widehat{U}, \widehat{Q}\rangle$. The simplest way to prove this is to introduce an auxilliary form of cellular automaton, the *contraction* cellular automaton, and first prove that any conventional CA can be simulated by a contraction automaton. Then we complete the proof by showing that any contraction automaton can be simulated by a 2^d-body cellular automaton.

Contraction automata are a straightforward generalization to higher dimensions of the *one-way* automata introduced by Culik [2]. Their sole function here is to systematically compress the neighborhood volume from the

$(2r+1)^d$ cells of a conventional CA with radius r to the 2^d cells available to a 2^d-body automaton, trading off [22] a reduction in the volume of the neighborhood for an increase in the size of the state set.

Definition 2.3 *A d-dimensional* contraction automaton *is a pair $\langle U, Q\rangle$ where Q is the state set and $U : Q^{2^d} \to Q$ is the local evolution operator,*

$$a^{t+1}_{i_1,i_2,\ldots,i_d} = U\begin{pmatrix} a^t_{i_1,i_2,i_3\ldots,i_d} \\ a^t_{i_1+1,i_2,i_3\ldots,i_d} \\ a^t_{i_1,i_2+1,i_3\ldots,i_d} \\ a^t_{i_1+1,i_2+1,i_3\ldots,i_d} \\ \vdots \\ a^t_{i_1+1,i_2+1,i_3+1\ldots,i_d+1} \end{pmatrix}. \qquad (2.7)$$

Each cell has a neighborhood consisting of the 2^d cells having coordinates equal to or one greater than the corresponding coordinate of the cell. For $d = 1, 2$ and 3, the neighborhoods for contraction automata are shown in figure 6.

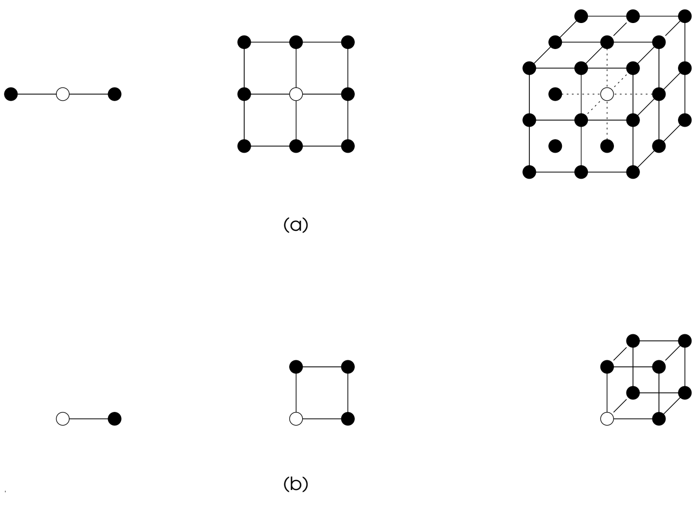

Figure 6: Comparison of the neighborhoods of the open cell in (a) conventional CA ($r = 1$ as shown) and (b) contraction automata for $d = 1, 2, 3$.

Lemma 2.1 *Any d-dimensional cellular automaton* $\langle U, Q, r\rangle$ *can be simulated by a d-dimensional contraction automaton* $\langle U', Q'\rangle$*. The simulation can be performed in the same amount of space with a slowdown of at most* $2r$ *and employs* $|Q'| = \sum_{j=1}^{2r} |Q|^{j^d}$ *states per cell.*

PROOF. For the sake of concreteness, we present the construction for the case $d = 2$ and $r = 1$; generalization to any $d \geq 1$ and $r \geq$ is straightforward.

First, Q' is enlarged to include the tensor products of Q needed to encode a contracted representation of the $(2r+1)^d$ cells in the conventional CA neighborhood.

$$Q' \equiv Q \cup (\otimes_{i=1}^{2^d} Q) \cup \ldots \cup (\otimes_{i=1}^{(2r)^d} Q) = \bigcup_{j=1}^{2r} Q^{j^d} \tag{2.8}$$

Since $r = 1$ and $d = 2$ in figure 7, the only new states introduced into Q' are $e \times f \times a \times b, f \times g \times b \times c, \ldots \in Q^4$. The product symbols will henceforth be omitted for the sake of brevity.

By construction, the local evolution rule U' has two distinct parts. First, it performs a sequence of contraction steps after which each cell is in a state encoding the states of all $(2r+1)^d$ cells of a neighborhood of $\langle U, Q, r\rangle$. In a single additional step, U' then simulates the action of U on that neighborhood.

For example, in the lower left corner of figure 7, the local evolution operator U' acts at time t_0 to produce the contracted state *efab* at time $t_0 + 1$. After U' has been applied everywhere, the neighbors of the corner cell are in the states *efab, fgbc, ijef* and *jkfg.* These cells contain enough information for U' to produce the contracted state *ijk*$\times$*efg*$\times$*abc*$\in Q^9$ which summarizes an entire 3×3 neighborhood of cell f at t_0. (That is, the accumulated neighborhood information is not centered around the cell originally in state a, but around the cell originally in state f.) But instead of recording the state of the neighborhood in an element of Q^9, we let U' be defined to simulate U on that neighborhood, e.g.,

$$U' \begin{pmatrix} efab \\ fgbc \\ ijef \\ jkfg \end{pmatrix} = U(f, e, g, i, j, k, a, b, c) = f'. \tag{2.9}$$

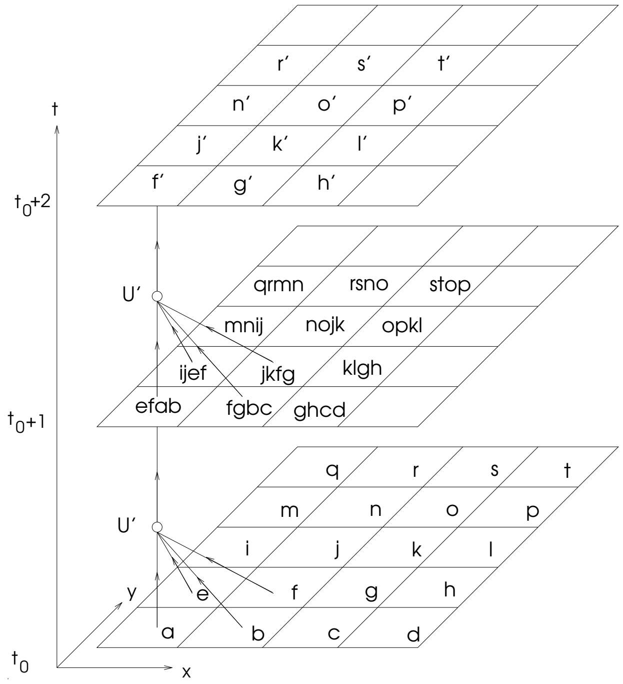

Figure 7: Simulation of an arbitrary two-dimensional cellular automaton by a two-dimensional contraction cellular automaton.

In the general case, $2r - 1$ applications of U' are required to contract the state information of the $(2r + 1)^d$ neighbors of a cell of $\langle U, Q, r\rangle$ into the 2^d-cell neighborhood of $\langle U', Q'\rangle$. Since an additional application of U' is required to simulate U on that neighborhood, the simulation proceeds at a rate $2r$ times slower than the conventional automaton being simulated. The net effect is that an exact simulation of $\langle U, Q, r\rangle$ is performed, except for a computationally unimportant drift of the simulated configuration with velocity $(-r, -r, \ldots - r)$ per simulated step. ∎

The principal cost of simulation manifests itself not in the slowdown, but in the number states $|Q'|$ required to simulate $\langle U, Q, r\rangle$. From eqn. 2.8, we see that $|Q'| = \sum_{j=1}^{2r} |Q|^{j^d}$. Unless both r and $|Q|$ are small integers, $|Q'|$, which is bounded below by

$$|Q'| > \int_0^r e^{x^d \ln|Q|} dx, \quad \text{when} \;\; d, r \geq 1 \;\; , |Q| \geq 2, \tag{2.10}$$

grows rapidly.

3 Simulation by N-body scattering automata

Theorem 3.1 *Any* d*-dimensional cellular automaton* $\langle U, Q, r\rangle$ *can be simulated by a 2-phase,* 2^d*-body automaton* $\langle \widehat{U}, \widehat{Q}\rangle$*. The* 2^d*-body automaton takes* $2r$ *steps to simulate each step of* $\langle U, Q, r\rangle$*, needs at most* 2^d *times as much space, and employs* $|\widehat{Q}| = \sum_{j=1}^{2r} |Q|^{j^d}$ *states.*

PROOF. As above, the construction is given explicitly for $d = 2$, but in such a way that the generalization to any $d \geq 1$ is clear. Since Lemma 2.1 assures us that for any $\langle U, Q, r\rangle$ there exists a contraction automaton $\langle U', Q'\rangle$ that simulates it, it suffices to first construct the 2^d-body cellular automaton $\langle \widehat{U}, \widehat{Q}\rangle$ from $\langle U', Q'\rangle$ and then show that $\langle \widehat{U}, \widehat{Q}\rangle$ can simulate any finite volume V of $\langle U', Q'\rangle$ using a volume no larger than $2^d V$.

Let $\widehat{Q} = Q'$. By definition 2.3, in $\langle U', Q'\rangle$ each cell belongs simultaneously to 2^d neighborhoods, whereas (by definition 2.2) in $\langle \widehat{U}, Q'\rangle$ each cell can belong to only a single neighborhood at any given time. Therefore, the initial state of $\langle \widehat{U}, Q'\rangle$ must contain at least 2^d copies of the initial state of $\langle U', Q'\rangle$ and so requires at least a volume $2^d V$ (see figure 8 for $d = 2$ and figure 9 for $d = 1$). Since the same argument holds at each subsequent step of the simulation of $\langle U', Q'\rangle$ by $\langle \widehat{U}, Q'\rangle$, the local evolution operator

$$\widehat{U} : (Q')^{2^d} \to (Q')^{2^d} \tag{3.1}$$

must be constructed to produce 2^d copies of the value U' would produce on the same input; that is,

$$\widehat{U}\begin{pmatrix} a^t_{x_1,x_2,x_3\ldots,x_d} \\ \vdots \\ a^t_{x_1+1,x_2+1,x_3+1\ldots,x_d+1} \end{pmatrix} \equiv \begin{pmatrix} U'\begin{pmatrix} a^t_{x_1,x_2,x_3\ldots,x_d} \\ \vdots \\ a^t_{x_1+1,x_2+1,x_3+1\ldots,x_d+1} \end{pmatrix} \\ \vdots \\ U'\begin{pmatrix} a^t_{x_1,x_2,x_3\ldots,x_d} \\ \vdots \\ a^t_{x_1+1,x_2+1,x_3+1\ldots,x_d+1} \end{pmatrix} \end{pmatrix}. \tag{3.2}$$

From figures 8 and 9, we see that in fact a volume $2^d V$ is both necessary and sufficient. As indicated by the dashed boxes in figure 9, the 2^d-fold copies of the simulated configuration of $\langle U', Q' \rangle$ shift at the rate of $(1/2, 1/2, ..., 1/2)$ per operation of $\widehat{U}$; after $2T$ steps of simulation, the simulated configuration must be read relative to the translated origin at $(T, T, ..., T)$.

Since Lemma 2.1 asserts that for any $\langle U, Q, r \rangle$ there exists a contraction automaton $\langle U', Q' \rangle$ that simulates it with slowdown $2r$ and $|Q'| = \sum_{j=1}^{2r} |Q|^{j^d}$, it follows that for any $\langle U, Q, r \rangle$ there exists a 2^d-body automaton $\langle \widehat{U}, Q' \rangle$ that simulates $\langle U, Q, r \rangle$ with the same number of states and the same slowdown in a volume 2^d-times larger. ∎

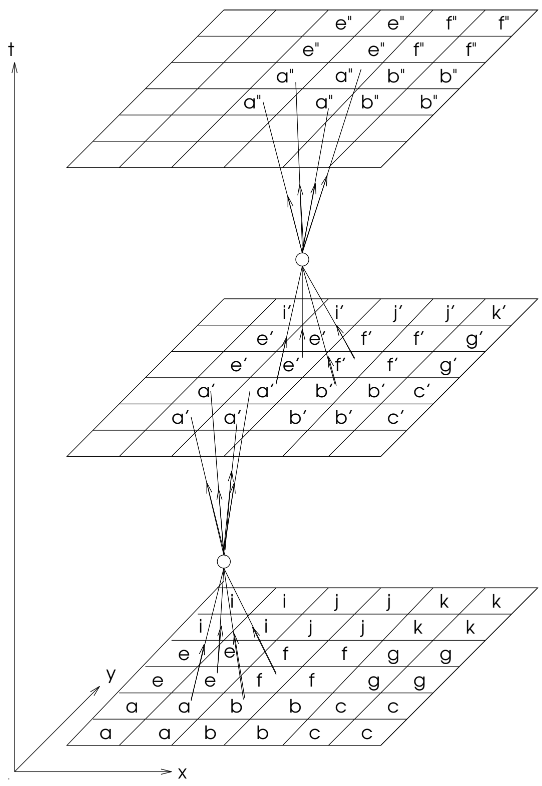

Figure 8: Simulation of a contraction automaton by a 2-phase, 2^d-body scattering automaton, for $d = 2$.

4 Small universal 2-body cellular automata

In the Introduction, we have emphasized the importance of finding numerical bounds on the minimum number of bits per cell required to perform universal computation in a few-body cellular automaton. Fabricators of cellular-automata device-arrays can use these bounds to assess the feasibility of using various physical couplings between device-cells without losing the potential for universal computation. Coupling mechanisms that are physically incapable of supporting enough bits per cell to be universal must be abandoned, or at best redirected toward special-purpose applications. Two of the most stringent cases, with clusters of size $N = 2$ and $N = 4$, occur for $d = 1, 2$ respectively ($N = 3$ does not occur on a square lattice, but can occur on a triangular lattice). If we apply Theorem 3.1 to conventional computation-universal CA having small neighborhood radii and small state sets, we can set an upper bound on the number of bits per cell required. For $d = 1$ ($N = 2$), figure 9 shows explicitly how the final stage of the simulation proceeds. The number of bits per cell required to simulate various conventional CA are discussed below and summarized in Tables 1 and 2.

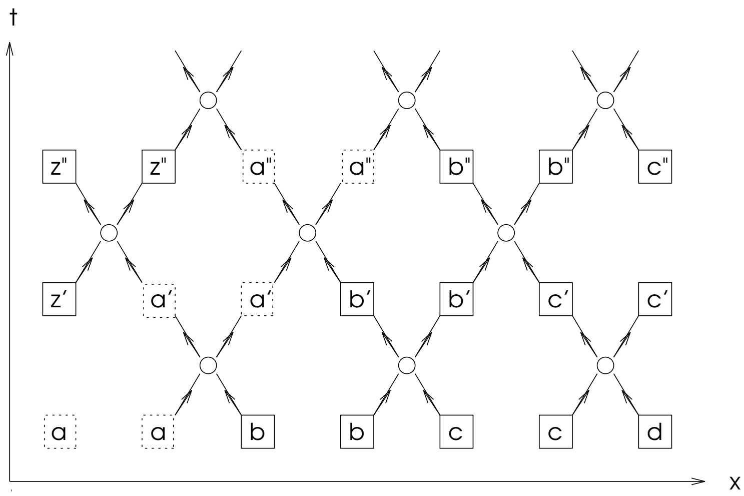

Figure 9: Simulation of a one-dimensional contraction (i.e., one-way) automaton by a 2-body automaton.

From the time von Neumann first introduced his two-dimensional, self-reproducing cellular automaton with 29 states, many researchers [8, 10, 3, 22] have attempted to reduce both the state count $|Q|$ and the neighborhood radius r of computation-universal rules. The earliest work focused on decreasing the number of states in von Neumann's automaton; Codd [10] reduced $|Q|$ from 29 to 8; Banks [3] later reduced $|Q|$ to 4. From Theorem 3.1, we see that by simulating the von Neumann, Codd and Banks constructions we obtain 4-body automata that are computation-universal and require $\log_2 |Q'| =$ 19.43, 12.00 and 8.02 bits per cell respectively.

Banks also discovered a rule in $d = 2$ which, like Conway's well-known "Life" rule, has the fewest possible states, $|Q| = 2$, the smallest possible radius, $r = 1$, but is still provably computation-universal. The 4-body cellular automata that simulate these rules each require just 3.32 bits per cell. Together with Margolus's one-bit version of Fredkin's universal billiard ball model [27] (which was, however, implemented from the start as a 4-body CA), the Banks and Conway rules represent the smallest possible conventional cellular automata and constitute the smallest known universal cellular automata in $d = 2$.

In contrast, for $d = 1$ the problem of minimizing $|Q|$ while retaining universality remains open. A. R. Smith[22, 23] has proven a series of theorems on the simulation of (m, n)-Turing machines[1] by one-dimensional conventional cellular automata. By applying his theorems to the small $(6, 6)$- and $(4, 7)$-Turing machines that Minsky [28] has shown to be universal, Smith has produced universal one-dimensional cellular automata with neighborhood radii $r = 1, 2$ and 3.

Theorem 4.1 (Smith [23]) *For any Turing machine with m symbols and n states, there exists a one-dimensional CA $\langle U, Q, r\rangle$ with $|Q| = \max(m, n) + 1$ and $r = 3$ that simulates it in real-time.*

For the $(6, 6)$-Turing machine of [28], this yields a conventional CA with $|Q| = 7$, $r = 3$. Applying Theorem 3.1 yields a 2-body CA with $|Q'| =$ $137,256$ or $\log_2 |Q'| = 17.07$ bits per cell.

Theorem 4.2 (Smith [23]) *For any Turing machine with m symbols and n states, there exists a one-dimensional CA $\langle U, Q, r\rangle$ with $|Q| = m + n$ and $r = 2$ that simulates it in real-time.*

[1] That is, Turing machines employing m tape symbols and n head states.

For the $(4, 7)$-Turing machine of [28], this yields a conventional CA with $|Q| = 11,\ r = 2$. Applying Theorem 3.1 yields a 2-body CA with $|Q'| = 16,104$ or $\log_2 |Q'| = 13.98$ bits per cell.

Theorem 4.3 (Smith [23]) *For any Turing machine with m symbols and n states, there exists a one-dimensional CA $\langle U, Q, r\rangle$ that simulates it in 2 times real-time and has $|Q| = m + 2n$ and $r = 1$.*

For the $(4, 7)$-Turing machine of [28], this yields a conventional CA with $|Q| = 18,\ r = 1$. Applying Theorem 3.1 yields a 2-body CA with $|Q'| = 342$ or $\log_2 |Q'| = 8.42$ bits per cell.

Culik at al. [2] have reduced the state set to $|Q| = 14$ with $r = 1$; the resulting universal CA yields a universal 2-body CA with $|Q'| = 210$ or $\log_2 |Q'| = 7.71$ bits per cell.

Recently, Lindgren et al. [26] have strengthened Theorems 4.2 and 4.3.

Theorem 4.4 (Lindgren and Nordahl [26]) *For any Turing machine with m symbols and n states, there exists a one-dimensional CA $\langle U, Q, r\rangle$ that simulates it in 2 times real-time and has $|Q| = m + n + 2$ and $r = 1$.*

Applying Theorem 4.4 to Minsky's $(4, 7)$-Turing machine would only reduce the state set to $|Q| = 13$, but Lindgren et al. reduce this further to $|Q| = 7$ and $r = 1$ by simulating the Turing machine head with composite objects propagating in a periodic background. Applying Theorem 3.1 to their construction yields a universal 2-body CA requiring just 5.81 bits per cell.

Smaller upper bounds on the required number of bits per cell in $d = 1$ have not yet been obtained, but several members of Wolfram class 4, widely conjectured to be universal, require fewer than 5.81 bits per cell. The two $r = 1$ rules, W^3_{357} and W^3_{824}, (where W^k_c denotes the k-ary rule with Wolfram [38] code c) both have $|Q| = 3$, and hence would yield $\log_2 |Q'| = 3.58$ bits per cell if their universality can be established. At the extreme limit, if a binary CA with $r = 1$, such as Wolfram rule 110 is universal[38, 26], the required amount of state could be reduced to 2.58 bits per cell.

Since we are primarily interested in minimizing N for universal N-body automata, we can obtain 2-body automata with even fewer states by applying the contraction operation within the 2×2 clusters of universal 4-body automata (see figure 10). It is not difficult to see that

(d=2) Rule	$\lvert Q\rvert$	r	Universal	$\log_2 \lvert Q'\rvert$
von Neumann [37, 8]	29	1	y	19.43
Codd [10]	8	1	y	12.00
Banks [3]	4	1	y	8.02
Conway("Life") [6]	2	1	y	3.32
Banks [3]	2	1	y	3.32

Table 1: Simulation of two-dimensional conventional automata by 4-body automata.

Claim 4.1 *Any 4-body automaton* $\langle \widehat{U}, \widehat{Q} \rangle$ *can be simulated by a 2-body automaton* $\langle \widetilde{U}, \widetilde{Q} \rangle$ *at the cost of a slowdown of* 2 *and augmentation of the state set* $|\widetilde{Q}| = 2|\widehat{Q}|$.

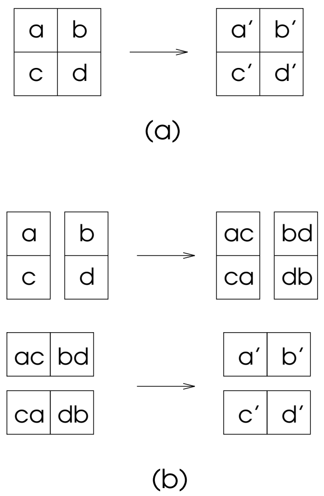

Figure 10: Simulation of a 4-body automaton by a 2-body automaton.

(d=1) Rule	$\lvert Q\rvert$	r	Universal	$\log_2 \lvert Q'\rvert$
Smith [23]	7	3	y	17.07
Smith [23]	11	2	y	13.98
Smith [23]	18	1	y	8.42
Lindgren et al. [26]	4	2	y	8.41
Albert et al. [2]	14	1	y	7.71
Lindgren et al. [26]	7	1	y	5.81
W^3_{357}, W^3_{824}	3	1	—	3.58
W^2_{110}	2	1	—	2.58

Table 2: Simulation of one-dimensional conventional automata by 2-body automata.

5 Induced complexity measure

We can also use Theorem 3.1 to define a simple measure of the complexity of a cellular automaton rule.

In the past, several mathematically plausible measures have been suggested (see [2] and references therein). Probably the most common approach [23], often used implicitly, has been to adopt the product $\mathcal{C}(r,k) = (r+1)^d k$ of the number of neighbors and the number of states $k \equiv |Q|$ as a measure of the complexity of $\langle U, Q, r\rangle$.

Since the 2-body automaton is perhaps the simplest form that physical computation can take, we adopt it as a "canonical form" for cellular automata; instead of trying to directly compare the complexity of rules with various $|Q|$ and r, we first apply the simulation theorem (Theorem 3.1) and then compare the complexity of the resulting canonical forms. But we still need an additional assumption to give us a physically plausible way to weigh the relative importance of r and $|Q|$. Following the Feynman-Toffoli conjecture [16, 33] that only a finite amount of computation can occur in a finite volume of space-time, we will assume that the spatial density of bits that are physically available to any simulator is some finite ρ.

For convenience we define the unit of volume to be $1/\rho$, so that each cell can then be used to represent a single bit of state; if more than one bit of state is required, composite cells must be constructed. If k distinct states are required, a configuration of $\lceil \log_2 k\rceil$ cells must be used to encode the k states.

If we now use this system to simulate an arbitrary cellular automaton, via the construction in Theorem 3.1, then the theorem tells us explicitly that we will need to use a spatial volume of at least $2^d \log_2 \left(\sum_{j=1}^{2r} |Q|^{j^d}\right)$ cells in order to have access to enough bits to represent the states required for the simulation. In addition, the simulation will take $2r$ times as long to run as the rule being simulated. So for example, the physical simulation of any (r, k)-rule in $d = 1$ consumes a volume of space-time that is a factor

$$\mathcal{D} = 4r \log_2 \left(\sum_{n=1}^{2r} k^n \right). \tag{5.1}$$

larger than that consumed by the original rule. We can take this space-time "dilution factor" $\mathcal{D}$ as a measure of the complexity $\mathcal{C}(r, k)$ of a cellular automaton rule.

If we approximate the sum by the corresponding integral, then we can write

$$\mathcal{D}(r, \ln k) = 4r \log_2 \left(\frac{e^{2r \ln k} - 1}{\ln k} \right) = 4r \log_2 \left(\frac{k^{2r} - 1}{\ln k} \right). \tag{5.2}$$

Further simplification of this expression is obtained by considering some reasonable limits. For $r \geq 1$ and $k \geq 2$, $k^{2r} \gg 1$ and $\ln(k^{2r} - 1) \approx 2r \ln k \gg \ln \ln k$; therefore,

$$\mathcal{D}(r, \ln k) = \frac{4}{\ln 2} r \ln \left(\frac{k^{2r} - 1}{\ln k} \right) \approx \frac{8}{\ln 2} r^2 \ln k = 8r^2 \log_2 k. \tag{5.3}$$

The graph of the exact expression $\mathcal{D}(r, \ln k) = 4r \log_2 \left(\sum_{n=1}^{2r} e^{n \ln k} \right)$ is plotted as a function of r and $\ln k$ in figure 11.

The approximation $\mathcal{D} \approx 8r^2 \log_2 k$ holds accurately even for small values of r and k. In addition, the measure $\mathcal{D}(r, \ln k)$ posesses a simple physical interpretation in terms of the Feynman-Toffoli conjecture.

6 Summary

We have attempted to find the minimum number of bits per cell required to perform universal computation in 2-body automata.

By applying Theorem 3.1 to universal conventional CA, we obtained an upper bound on the number of bits per cell required for universal computation

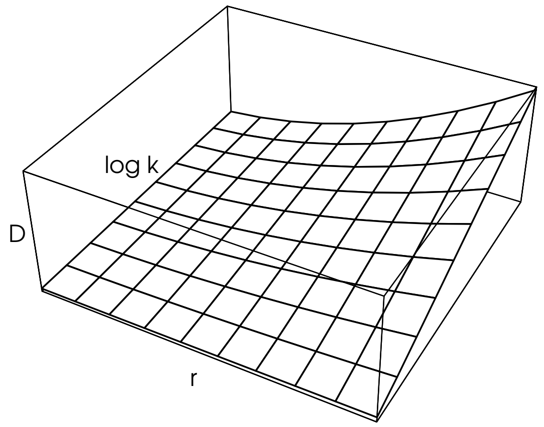

Figure 11: Plot of the exact expression, eqn. (5.1), for the dilution function, $\mathcal{D}(r, \log_2 k)$, with $r \in (1, 10)$ and $\log_2(k) \in (0, 9)$.

when only two-body interactions are permitted. The smalles upper bound we have established, based on a conventional CA constructed by Lindgren [26], is 5.81 bits per cell. If the hypothesized universality [38] of Wolfram rules 357 and 824 can be proven, the upper bound can immediately be reduced to 3.58 bits per cell. If any binary CA with $r = 1$ (such as rule 110) can be shown to be universal (and there is some disagreement about the possibility; compare [26] with [38, p.31]) then the required amount of state can be reduced to 2.58 bits per cell.

In $d = 2$ we have shown how to construct a computation-universal 2-body automaton requiring 2 bits per cell. This requirement can probably be met by many of the proposed schemes for computing with nanometer-scale arrays. The construction of a computation-universal 2-body automaton that only requires a single bit per cell remains an open problem.

The search for compact, yet computation-universal, cellular automata rules has intrigued researchers since the time of von Neumann. With the advent of dense cellular-automaton-like device arrays, the search may one day take on practical importance as well.

References

[1] N. W. Ashcroft and N. David Mermin, *Solid State Physics* (1976).

[2] J. Albert and Karel Culik, "A Simple Universal Cellular Automaton and its One-Way and Totalistic Versions", *Complex Systems*, **1** (1987) 1–16.

[3] E. R. Banks, "Information Processing and Transmission in Cellular Automata" (Doctoral Dissertation, Massachusetts Institute of Technology, 1971).

[4] R. T. Bate, "Nanoelectronics", *Solid State Technology* (November 1989), 101–108.

[5] R. T. Bate, G. A. Frazier, W. R. Frensley, J. W. Lee and M. A. Reed, "Prospects for quantum integrated circuits", in *Quantum Well and Superlattice Physics*, edited by Gottfreid H. Dohler (SPIE, International Society for Optical Engineering, 1987).

[6] E. R. Berlekamp and J. H. Conway and R. K. Guy, "Winning Ways for Your Mathematical Plays" (Academic Press, 1982).

[7] Michael Biafore, "Cellular Automata for Nanoscale Computation", *Physica D* (to appear).

[8] A. W. Burks, "Essays on Cellular Automata", (University of Illinois, 1970).

[9] F. L. Carter, "The Molecular Device Computer: Point of Departure for Large Scale Cellular Automata", *Physica D* **10** (1984) 175–194.

[10] E. F. Codd, "Cellular Automata" (Academic Press, 1968).

[11] R. H. Dennard, "Physical Limits to VLSI Technology Using Silicon MOSFETS", *Physica B* **117** (1983) 39–43.

[12] L. D. Faddeev, "Mathematical Aspects of the Three-body Problem in Quantum Scattering Theory", Israel Program for Scientific Translations (1965).

[13] D. K. Ferry, "Two-Dimensional Automata in VLSI", in *Submicron Integrated Circuits*, edited by R. K. Watts (Wiley-Interscience, 1988).

[14] G. D. Doolen, *Lattice Gas Methods for Partial Differential Equations* (North-Holland, 1989).

[15] D. K. Ferry and W. Porod, "Interconnections and Architecture for Ensembles of Microstructures", *Superlattices and Microstructures* **2** (1986) 41.

[16] R. P. Feynman, "Quantum Mechanical Computers", *Foundations of Physics* **16** (1986) 507–531.

[17] U. Frisch, B. Hasslacher and Yves Pomeau, "Lattice-Gas Automata for the Navier-Stokes Equation", *Physical Review Letters* **56** (1986) 1505–1508.

[18] J. Hardy, O. De Pazzis and Yves Pomeau, "Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions", *Physical Review A* **13** (1976) 1949–1960.

[19] J. Hardy, Yves Pomeau and O. De Pazzis, "Time evolution of a two-dimensional model system. I. Invariant states and time correlation functions", *Journal of Mathematical Physics* **14** (1973) 1746–1759.

[20] K. Hepp "On the Quantum Mechanical N-body Problem", *Helvetica Physica Acta* **42** (1969) 425–458.

[21] K. K. Likharev, "Correlated discrete transfer of single electrons in ultrasmall tunnel junctions", *IBM Journal of Research and Development* **32** (1988) 144–157.

[22] A. R. Smith III, "Cellular Automata Complexity Trade-Offs", *Information and Control* **18** (1971) 466–482.

[23] A. R. Smith III, "Simple computation-universal cellular spaces", *Journal of the ACM (Association for Computing Machinery)* **18** (1971) 339–353.

[24] J. D. Jackson, *Classical Electrodynamics* (Wiley, 1975).

[25] J. L. Prince, *Very Large Scale Integration* (Springer, 1980).

[26] K. Lindgren and M. Nordahl, "Universal Computation in Simple One-Dimensional Cellular Automata", *Complex Systems* **4** (1990)299–318.

[27] Norman Margolus, "Physics-like models of computation", *Physica D* **10** (1984) 81–95.

[28] Marvin Minsky, *Computation: Finite and Infinite Machines* (Prentice-Hall, 1967).

[29] J. A. Morton, "From Physics to Function", *IEEE Spectrum* (September 1963) 62–66.

[30] K. Obermayer and W. Teich and G. Mahler, "Structural basis of multistationary quantum systems. I. Effective single particle dynamics", *Physical Review B* **37** (1988) 8111–8121.

[31] R. T. Bate, "The future of microstructure technology", *Superlattices and Microstructures* **2** (November 1986) 9–11.

[32] J. R. Taylor, *Scattering Theory*, Academic Press (1972).

[33] T. Toffoli, "How Cheap Can Mechanics' First Principles Be?", in *Complexity, Entropy and the Physics of Information*, edited by W. Zurek (Santa Fe Institute, 1990).

[34] T. Toffoli, CAM: a high-performance cellular-automaton machine, *Physica* **10D** (1984) 195–205.

[35] T. Toffoli and N. Margolus, *Cellular Automata Machines* (MIT Press, 1987).

[36] N. Margolus and T. Toffoli, Cellular Automata Machines, in *Lattice Gas Methods for Partial Differential Equations*, eds. G. Doolen et al., Addison-Wesley (1990).

[37] J. von Neumann "Theory of Self-Reproducing Cellular Automata", (University of Illinois, 1966).

[38] Stephen Wolfram, "Universality and Complexity in Cellular Automata", *Physica D* **10** (1984) 1–35.