Complex Systems **7** (1993) 221–239

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 developed specifically to investigate the possibility of implementing a new generation of cellular automata machines that would use dense arrays of nanometer-scale device-cells. In this paper, we try to determine how many states per cell are required by few-body automata in order to perform universal computation. We prove theorems describing the space, time, and state-set complexity of the simulation of d-dimensional conventional cellular automata with N-body automata, and show that there exist computation-universal 2-body automata requiring 5.81 bits of state per cell for $d = 1$ and 2 bits per cell for $d = 2$. These results suggest that physically-imposed restrictions on the number of available bits per cell will not be an obstacle to cellular automaton-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, 15, 30, 31] have proposed an alternative architecture based on cellular automata (CAs). Specifically, these physicists envision dense arrays of nanometer-scale devices communicating with one another via *direct physical interactions* rather than through wired interconnections. The devices might consist of semiconductor heterostructures, molecular switches, or other nanometer-scale structures 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 CA in which each device acts as a single cell.

Researchers have proposed that devices be coupled via resonant quantum tunneling [4, 13, 31], Coulomb interaction effects [21, 30], or soliton switching in π-conjugated polyenes [9]. If such proposals were ever realized, the resulting CA machines would be unlike any that now exist [34, 35, 36]. The dynamics of the CA would no longer be determined by an arbitrarily

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

definable lookup table; instead, it would be completely determined by the particular physical interactions between cells [29].

Few-body automata [7] are a class of CAs developed to investigate the computational feasibility of such proposals. The geometric structure of a few-body automaton is defined so that it reflects 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 [14, 17, 18] or partitioning [27] CAs. 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 establishing 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 the most accessible experimentally and the most tractable theoretically. Second, the case of $N = 2$, where each cell has only a single neighbor aside from itself, is the most difficult limit (from a computational perspective) in which to achieve computation universality. Therefore, any upper bounds we 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 specific nanometer-scale devices as cells of a CA. The underlying physics limits some of the proposed coupling mechanisms (such as single-electron tunneling[21]) 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 2-body CAs are capable of universal computation, the relevant question 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 CA. By applying this simulation theorem to conventional CA that are known to be computation-universal, we establish the existence of a universal 2-body automaton requiring 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 computation-universal 2-body automaton requiring 2 bits per cell.

2. *N*-body CAs

To begin, we define explicitly what we will mean by "conventional" (or "von Neumann") CAs, and explain why they are not a satisfactory tool for analyzing the CA-like properties of arrays of interacting devices.

Conventional CAs [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

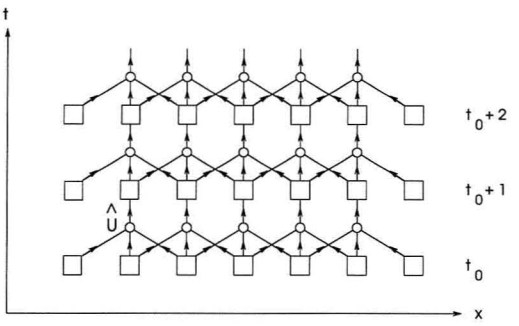

Figure 1: Time evolution of a conventional CA in one dimension.

evolution operator $\hat{U}$ with domain $Q^{(2r+1)^d}$ and range Q, where Q is the finite set of states each cell may occupy.

Definition 2.1. *On a d-dimensional rectangular lattice, a* conventional CA *with radius r is a triple $\langle \hat{U}, Q, r \rangle$, where Q is the finite set of states that each cell can occupy, r is the radius of the neighborhood and $\hat{U} : Q^{(2r+1)^d} \to Q$ 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 $(i_1, i_2, \ldots, i_d)$ depends on its own state at time t and those of neighboring cells, according to

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

The spacetime diagram for a one-dimensional conventional CA with $r = 1$ is shown in Figure 1.

Unfortunately, conventional CAs are not a suitable model for the application proposed by device physicists, in which cells are coupled by real physical interactions (such as Coulomb dipole interactions [7, 30]). Note that CAs are characterized by three discreteness properties:

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

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

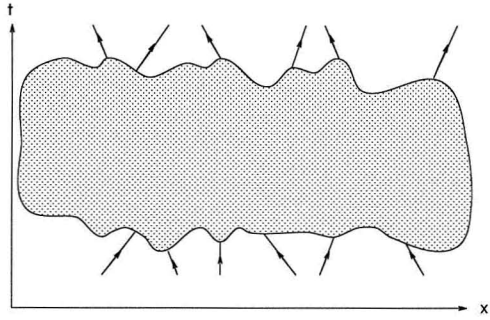

Figure 2: Generic process with six input states and six output states.

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

If we attempt to construct a conventional CA by simply arranging nanometer-scale devices in a rectangular array, then it becomes extremely difficult to retain properties 2 and 3. Because the cells are coupled by physical interactions, state transitions occur continuously rather than at controlled, discrete time-intervals. Likewise, although the effect of distant cells decreases with separation (for example, 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 it has little in common with CAs (other than its spatial discreteness). Until we find a format to constrain the dynamics of such device arrays so that they have all three forms of discreteness essential to CAs, we will not be able to analyze their computational properties within the framework of CAs theory.

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

Since we want to use physical interactions between cells to implement the CA, our approach is to let the form of the interactions determine the form of the CA, instead of adopting the definition of the conventional CA. 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.

If an interaction is not too singular when the interacting particles approach one another, and if it subsides quickly 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

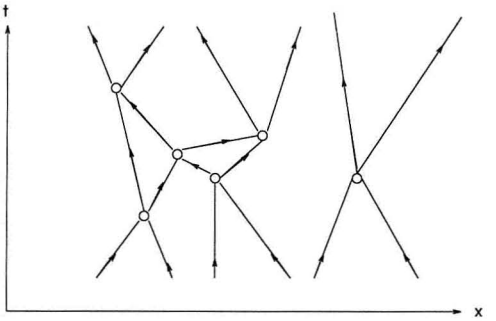

Figure 3: Description of a regular physical interaction process by an N-body diagram.

a scalar 2-body interaction is regular if it satisfies

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

and

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

Regular interactions are much more amenable to theoretical analysis, because the diagrammatic and algebraic tools of many-body theory can then be applied. By restricting our attention to those nanometer-scale devices that have regular interactions, we can base our definition of few-body automata on physical processes that take the form shown in Figure 3.

Nevertheless, the spacetime structure in Figure 3 is still too irregular in space, time, and number of interacting entities to be regarded as a CA. If we discretize the times and places at which interactions take place, however, we obtain the uniform spacetime structure shown in Figure 4—that of a one-dimensional, 2-body automaton. This should be contrasted with the spacetime diagram of the one-dimensional conventional CA shown in 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 its left or right neighbor. All 2-body automata have this spacetime structure, only the particular rule $\widehat{U} : (a_i^t, a_{i+1}^t) \rightarrow (a_i^{t+1}, a_{i+1}^t)$ which maps 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 our simulation theorem. Very general definitions of few-body automata are possible, because one can vary several 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

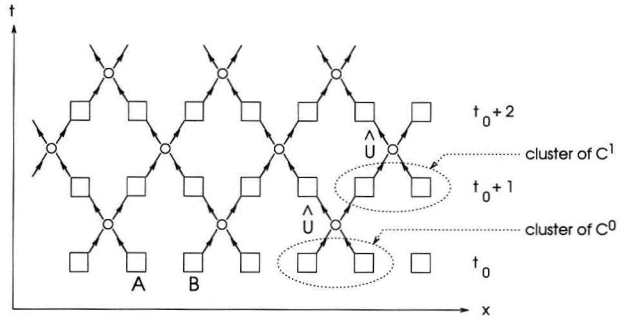

Figure 4: The time evolution of a 2-body CA in one dimension is identical to that of a one-dimensional lattice gas.

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; that is,

$$\mathcal{C}^0 = \{\mathcal{C}_{2i_1,2i_2,\ldots,2i_d} | i_k \in Z\}, \tag{4}$$

and

$$\mathcal{C}^1 = \{\mathcal{C}_{2i_1+1,2i_2+1,\ldots,2i_d+1} | i_k \in Z\}, \tag{5}$$

where

$$\mathcal{C}_{i_1,i_2,\ldots,i_d} = \{i_1+\tau_1, i_2+\tau_2, \ldots, i_d+\tau_d | i_k \in Z, \tau_k \in \{0,1\}\} \tag{6}$$

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 $\mathcal{C}^0$. At each time step, the rule $\hat{U}$ maps the state of all 2^d cells of a cluster into a new state for all 2^d cells. In general, the new state of each cell depends on the previous state of all cells in the cluster.

Definition 2.2. *A d-dimensional 2^d-body CA is a pair $\langle \hat{U}, \hat{Q} \rangle$, where $\hat{Q}$ is the finite set of states that each cell can occupy, and the automorphism $\hat{U} : \hat{Q}^{2^d} \to \hat{Q}^{2^d}$ 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.

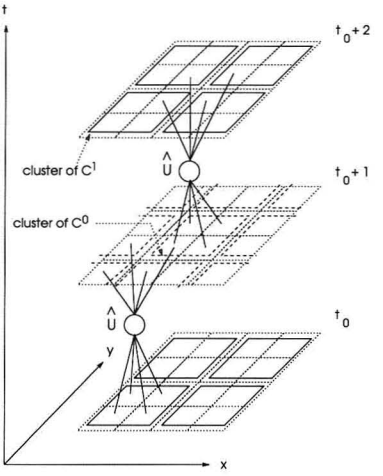

Figure 5: Four-body CA with 2×2 cluster neighborhoods. Clusters of $\mathcal{C}^0$ are shown as dashed squares; clusters of $\mathcal{C}^1$ are solid; the cells are shown as dotted boxes.

2.1 Simulation by contraction CAs

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

Contraction CAs are a straightforward generalization to higher dimensions of the *one-way* automata introduced by Culik [2]. Their sole function here is to systematically compress 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 CA, trading off an increase in the size of the state set for a reduction in the volume of the neighborhood [22].

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

$$a^{t+1}_{i_1,i_2,\ldots,i_d} = \hat{U} \begin{pmatrix} a^t_{i_1,i_2,i_3,\ldots,i_d} \\ a^t_{i_1+1,i_2,i_3,\ldots,i_d} \\ a^t_{i_1,i_2+1,i_3,\ldots,i_d} \\ a^t_{i_1+1,i_2+1,i_3,\ldots,i_d} \\ \vdots \\ a^t_{i_1+1,i_2+1,i_3+1,\ldots,i_d+1} \end{pmatrix}. \tag{7}$$

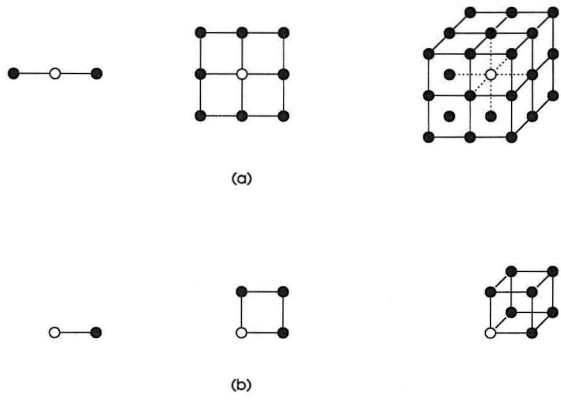

Figure 6: Comparison of the neighborhoods of the open cell in (a) conventional CAs ($r = 1$ as shown), and (b) contraction CAs for $d = 1, 2, 3$.

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 CAs are contrasted with those for conventional CAs in Figure 6.

Lemma 2.1. *Any d-dimensional CA $\langle U, Q, r\rangle$ can be simulated by a d-dimensional contraction cellular 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.*

For 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.

$$Q' \equiv Q \cup (\otimes_{i=1}^{2^d} Q) \cup \ldots \cup (\otimes_{i=1}^{(2r)^d} Q) = \bigcup_{j=1}^{2r} Q^{j^d} \tag{8}$$

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.

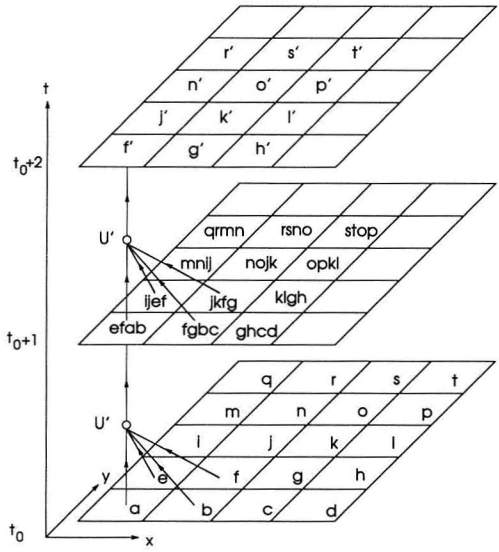

Figure 7: Simulation of an arbitrary two-dimensional CA by a two-dimensional contraction CA.

After U' has been applied to all neighborhoods, 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.,

$$U'\begin{pmatrix} efab \\ fgbc \\ ijef \\ jkfg \end{pmatrix} = U(f,e,g,i,j,k,a,b,c) = f'. \tag{9}$$

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 CA 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 of states $|Q'|$ required to simulate $\langle U, Q, r\rangle$. From (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

$$|Q'| > \int_0^r e^{x^d \ln |Q|} dx, \qquad \text{when } d, r \geq 1,\ |Q| \geq 2, \tag{10}$$

grows rapidly.

3. Simulation by N-body scattering automata

Theorem 3.1. *Any d-dimensional CA $\langle U, Q, r\rangle$ can be simulated by a 2-phase, 2^d-body cellular automaton $\langle \widehat{U}, \widehat{Q}\rangle$. The 2^d-body cellular 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.*

As in the previous section, the construction is given explicitly for $d = 2$, such that the generalization to any $d \geq 1$ is clear. Because Lemma 2.1 assures us that there exists a contraction CA $\langle U', Q'\rangle$ that simulates any CA $\langle U, Q, r\rangle$, it suffices to first construct the 2^d-body CA $\langle \widehat{U}, \widehat{Q}\rangle$ 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, each cell in $\langle U', Q'\rangle$ belongs simultaneously to 2^d neighborhoods, whereas (by definition 2.2) each cell in $\langle \widehat{U}, Q'\rangle$ 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 a volume of at least $2^d V$ (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

$$\widehat{U} : (Q')^{2^d} \to (Q')^{2^d} \tag{11}$$

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

$$\widehat{U}\begin{pmatrix} a^t_{x_1,x_2,x_3,\ldots,x_d} \\ \vdots \\ a^t_{x_1+1,x_2+1,x_3+1,\ldots,x_d+1} \end{pmatrix} \equiv \begin{pmatrix} U'\begin{pmatrix} a^t_{x_1,x_2,x_3,\ldots,x_d} \\ \vdots \\ a^t_{x_1+1,x_2+1,x_3+1,\ldots,x_d+1} \end{pmatrix} \\ \vdots \\ U'\begin{pmatrix} a^t_{x_1,x_2,x_3,\ldots,x_d} \\ \vdots \\ a^t_{x_1+1,x_2+1,x_3+1,\ldots,x_d+1} \end{pmatrix} \end{pmatrix}. \tag{12}$$

From Figures 8 and 9, we see that in fact a volume of $2^d V$ 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 $(1/2, 1/2, \ldots, 1/2)$ per operation of $\widehat{U}$; after $2T$ steps of simulation, the

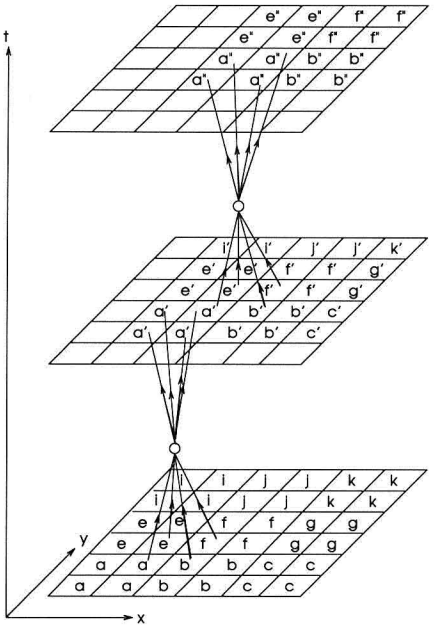

Figure 8: Simulation of a contraction CA by a 2-phase, 2^d-body scattering CA, for $d = 2$.

simulated configuration must be read relative to the translated origin at $(T, T, \ldots, T)$.

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

4. Small universal 2-body CAs

In the introduction to this paper, we emphasized the importance of finding numerical bounds on the minimum number of bits per cell required to perform universal computation in a few-body CA. Fabricators of CA 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 sizes $N = 2$ and $N = 4$, occur for $d = 1$ and 2, respectively ($N = 3$ does not occur on a square lattice). If we apply Theorem 3.1 to conventional computation-universal CAs having small neighborhood radii and small state

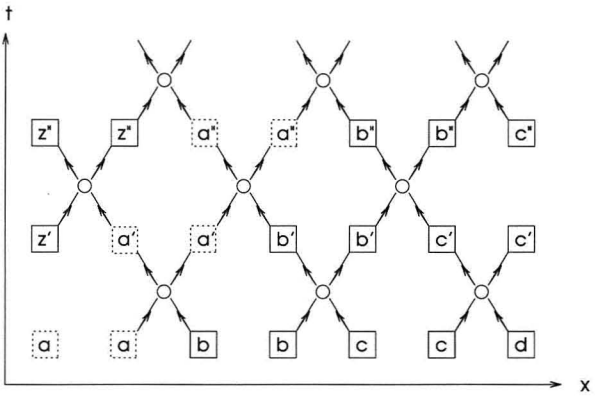

Figure 9: Simulation of a one-dimensional contraction (that is, one-way) CA by a 2-body CA.

sets, we can set an upper bound on the number of bits per cell required. For $d = 1$ ($N = 2$), Figure 9 explicitly shows how the final stage of the simulation proceeds. The numbers of bits per cell required to simulate various conventional CAs are discussed subsequently and summarized in Tables 1 and 2.

From the time von Neumann first introduced his two-dimensional, self-reproducing CA with 29 states, researchers have attempted to reduce both the state count $|Q|$ and the neighborhood radius r of computation-universal rules [3, 8, 10, 22]. The earliest work focused on decreasing the number of states in von Neumann's CA; Codd [10] reduced $|Q|$ from 29 to 8; Banks [3] later reduced $|Q|$ to 4. Theorem 3.1, implies 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 $d = 2$ that, like Conway's well-known "Life" rule, has the fewest possible states, $|Q| = 2$, the smallest possible radius, $r = 1$, but is still demonstrably computation-universal. The 4-body CAs 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 (which was, however, implemented from the start as a 4-body CA) [27], the Banks and Conway rules represent the smallest possible conventional CAs and constitute the smallest known universal CAs in $d = 2$.

In contrast, the problem of minimizing $|Q|$ while retaining universality for $d = 1$ 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 CAs. By applying his theorems to the small $(6, 6)$- and $(4, 7)$-Turing machines

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

$(d = 2)$ Rule	$\|Q\|$	r	Universal	$\log_2 \|Q'\|$
von Neumann [8, 37]	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 CAs by 4-body CAs.

$(d = 1)$ Rule	$\|Q\|$	r	Universal	$\log_2 \|Q'\|$
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 CAs by 2-body CAs.

that Minsky [28] has shown to be universal, Smith has produced universal one-dimensional CAs 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], Theorem 4.1 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.*

For the $(4, 7)$-Turing machine of [28], Theorem 4.2 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], Theorem 4.3 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 et 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 and Nordahl [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 the (4, 7)-Turing machine of [28] would only reduce the state set to $|Q| = 13$, but Lindgren and Nordahl 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 code c [38]), both have $|Q| = 3$, and would therefore yield $\log_2 |Q'| = 3.58$ bits per cell if their universality could be established. At the extreme limit, if a binary CA with $r = 1$, such as Wolfram rule 110, is universal [26, 38], the amount could be reduced to 2.58 bits per cell.

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

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

5. Induced complexity measure

We can also use Theorem 3.1 to define a simple measure of the complexity of a CA rule. Several mathematically plausible measures have been suggested previously (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 $k \equiv |Q|$ of states) as a measure of the complexity of $\langle U, Q, r\rangle$.

Since the 2-body CA is perhaps the simplest form that physical computation can take, we adopt it as a "canonical form" for CAs; 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

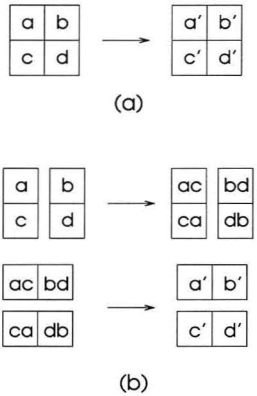

Figure 10: Simulation of a 4-body CA by a 2-body CA.

the resulting canonical forms. However, we still need an additional assumption to give us a physically plausible means to weigh the relative importance of r and $|Q|$. Following the Feynman-Toffoli conjecture that only a finite amount of computation can occur in a finite volume of space-time [16, 33], 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 use this system to simulate an arbitrary CA, via the construction in Theorem 3.1, a spatial volume of at least $2^d \log_2 \left(\sum_{j=1}^{2r} |Q|^{j^d} \right)$ cells will be required to have access to enough bits to represent the states for the simulation. In addition, the simulation will take $2r$ times as long to run as the rule being simulated. Thus, the physical simulation of any (r, k)-rule in $d = 1$, for example consumes a volume of space-time that is a factor

$$\mathcal{D} = 4r \log_2 \left(\sum_{n=1}^{2r} k^n \right) \tag{13}$$

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 CA rule.

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

$$\mathcal{D}(r, \ln k) = 4r \log_2 \left(\frac{e^{2r \ln k} - 1}{\ln k} \right) = 4r \log_2 \left(\frac{k^{2r} - 1}{\ln k} \right). \tag{14}$$

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 $\ln(k^{2r} - 1) \approx 2r \ln k \gg$

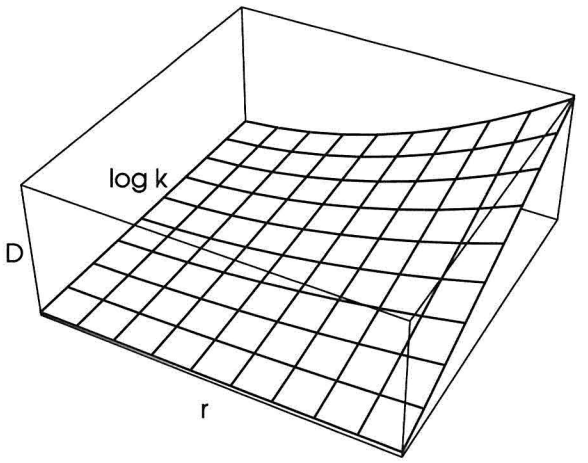

Figure 11: Plot of the exact expression (13), for the dilution function $\mathcal{D}(r, \log_2 k)$, with $r \in (1, 10)$ and $\log_2(k) \in (0, 9)$.

$\ln \ln k$; therefore,

$$\mathcal{D}(r, \ln k) = \frac{4}{\ln 2} r \ln \left(\frac{k^{2r} - 1}{\ln k} \right) \approx \frac{8}{\ln 2} r^2 \ln k = 8r^2 \log_2 k. \tag{15}$$

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

6. Summary

By applying Theorem 3.1 to universal conventional CAs, we have obtained an upper bound on the number of bits per cell required for universal computation when only two-body interactions are permitted. The smallest upper bound that we have established, based on a conventional CA constructed by Lindgren and Nordahl [26], is 5.81 bits per cell. If the hypothesized universality of Wolfram rules 357 and 824 [38] 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 (there is some debate about this possibility; compare [26] with [38, p.31]), then the upper bound can be reduced to 2.58 bits per cell.

For $d = 2$, we have shown how to construct a computation-universal 2-body CA that requires 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 CA that requires only a single bit per cell remains an open problem.

The search for compact yet computation-universal, CA rules has provided an intriguing theoretical challenge since von Neumann's pioneering contribution [37]. With the advent of dense CA-like device arrays, the search may soon take on practical importance as well.

7. Acknowledgments

I would like to thank the Advanced Research Projects Agency and the Office of Naval Research for supporting this research under Ultra Project grant N00014-93-1-0660.

References

[1] N. W. Ashcroft and N. David Mermin, *Solid State Physics* (Philadelphia: Saunders/HRW, 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" unpublished doctoral dissertation, Massachusetts Institute of Technology (1971).

[4] R. T. Bate, "Nanoelectronics," *Solid State Technology*, **32** (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," pages 26–34 in *Quantum Well and Superlattice Physics*, edited by Gottfreid H. Dohler (SPIE, International Society for Optical Engineering, 1987).

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

[7] Michael Biafore, "Cellular Automata for Nanoscale Computation," *Physica D*, forthcoming.

[8] A. W. Burks, *Essays on Cellular Automata* (Urbana: 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* (Cambridge, MA: 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," pages 377–413 in *Submicron Integrated Circuits*, edited by R. K. Watts (New York: Wiley, 1988).

[14] G. D. Doolen, *Lattice Gas Methods for Partial Differential Equations* (Amsterdam: 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*, **18** (1971) 339–353.

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

[25] J. L. Prince, *Very Large Scale Integration* (New York: 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* (Englewood Cliffs, NJ: Prentice-Hall, 1967).

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

[30] K. Obermayer, 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** (Nov. 1986) 9–11.

[32] J. R. Taylor, *Scattering Theory*, (Cambridge, MA: 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, NM: Santa Fe Institute, 1990).

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

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

[36] N. Margolus and T. Toffoli, "Cellular Automata Machines," pages 219–248 in *Lattice Gas Methods for Partial Differential Equations*, edited by G. Doolen et al., (Reading, MA: Addison-Wesley, 1990).

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

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