Notes

A Note on Cellular Automata Simulations

JON T. BUTLER*

Dept. of Electrical Engineering, The Technological Institute,
Northwestern University, Evanston, Illinois 60201

During the past several years, a number of papers have appeared on the subject of simulation in cellular automata systems. This paper describes a simulation of a two-dimensional system using a square neighborhood of \( n \times n \) cells by one, using the generalized von Neumann neighborhood, which costs significantly less in terms of cell complexity than existing simulations. In addition, the equivalences between certain neighborhoods are demonstrated.

1. INTRODUCTION

A cellular automata system or cellular space in two dimensions can be viewed as a connection of finite state machines, each situated on a square in an infinitely large sheet of graph paper. Each machine receives an input from neighboring machines, and sends, in turn, its output to neighboring machines. Thus, the state of each machine at time \( t \) depends on its state at \( t - 1 \) and on the states of its neighbors at \( t - 1 \).

The cellular space was first used to demonstrate a system capable of self-reproduction and universal computation (von Neumann, 1966). Because of the planar arrangement of cells, a cellular array is well suited for picture processing, and several papers deal with this aspect (Unger, 1958; Unger, 1959). In the recent past, other papers have treated the subject of equivalences between cellular spaces (Cole, 1969; Yamada and Amoroso, 1971; Smith, 1971; Hamacher, 1971; Amoroso and Guilfoyle, 1972). Such studies have been motivated by the fact that, given certain spaces, other spaces can be found which are more optimum in terms of speed or complexity. In general, a tradeoff is involved. For example, given one space, another space may be found with a shorter computation time, but with a more complex cell. The

*Partially supported by the National Research Council under a Postdoctoral Associateship and by an IBM Fellowship, Ohio State University.
value of the tradeoff, in this case, can be measured by the amount of increase in cell complexity, for some specified decrease in computation time.

This paper describes a simulation of a two-dimensional cellular space using a square neighborhood of \( n \times n \) cells by one, using the generalized von Neumann neighborhood which costs significantly less in terms of cell complexity than existing simulations.

The definition of the cellular space and simulation is presented in Section 2 below. Section 3 demonstrates equivalences between certain cell neighborhoods and presents the improved simulation.

### 2. Definitions and Notation

In this section, the concept of a cellular array is presented, together with the idea of a simulation of one array by another. The definitions and notation conform closely to that of Smith (1971).

A cellular array exists on a \( d \)-dimensional \((d \rightarrow D)\) lattice of points indexed by \( \mathbb{Z}^d \), the set of \( d \)-tuples of integers. Each point \( \bar{a} \) in the lattice is uniquely specified as \( \bar{a} = (a_0, a_1, a_2, ..., a_{d-1}) \). The special point, \( \bar{0} \), is called the origin and has all 0 components, i.e., \( \bar{0} = (0, 0, ..., 0) \).

At each lattice point, \( \bar{m} \in \mathbb{Z} \), is situated a Moore finite-state machine \( G \). The exact copy of \( G \) at \( \bar{m} \) is \( A_{\bar{m}} \), a cell in the array. \( G \) is defined in the usual way as \( G = (X, Y, Q, f, \beta) \), where \( X \) is a set of inputs, \( Y \) is a set of outputs, \( Q \) is a set of internal states, and \( f \) is the next state mapping—it specifies the next state of \( G \) as a function of the present state and present inputs. Thus, \( f: Q \times X \to Q \), \( \beta \) is the output mapping; it specifies the output of \( G \) as a function of the present state. Thus, \( \beta: Q \to Y \).

In a cellular array, each machine has \( n \) input lines which connect to the outputs of \( n \) machines at neighboring lattice points. The inputs convey the states of corresponding machines, and so \( X = Q \times Q \times \cdots \times Q = Q^n \). Let \( N = (\bar{a}_1, \bar{a}_2, ..., \bar{a}_n) \), the neighborhood index, be an ordered subset of points in \( \mathbb{Z}^d \), such that \( \bar{a}_i = \bar{0} \). The cell which supplies the \( i \)th input to a machine \( G \) at \( \bar{x} \) is specified by \( \bar{a}_i + \bar{x} \), and the set of \( n \) cells is the neighborhood of \( \bar{x} \). Similarly, each \( G \) provides an output to all machines in the neighborhoods to which it belongs, and so \( Y = Q \times Q \times \cdots \times Q = Q^n \). The output mapping is restricted so that all output lines from each \( G \) provide the same state to each cell; thus, \( \beta(q) = (q, q, ..., q) \).

One state \( q_0 \in Q \), the quiescent state, has the property that a machine in \( q_0 \) at time \( t \) will remain in \( q_0 \) at \( t + 1 \), if all the machines which supply inputs to it are also in \( q_0 \) at \( t \). Thus, \( f(q_0; q_0, ..., q_0) = q_0 \).
A uniform cellular space \( Z \) (or simply cellular space) is a triple \((G, d, N)\), in which \( G \), the Moore machine is the same for all points in \( \mathbb{Z}^d \) and \( N \), the neighborhood index, is the same for all machines.

In addition to the neighborhood index, it is convenient to introduce a similar concept. Formally, a template \( T \) in a cellular space \( Z \) with neighborhood index \( N = (\vec{a}_1, \vec{a}_2, ..., \vec{a}_n) \) is the unordered set \( T = \{d_1, d_2, ..., d_n\} \). The template origin is \( \vec{a}_1 \) \((= \vec{0})\).

Four templates are of special interest in this paper. The first template has the following form:

\[
H_k^{(d)} = \{ \vec{a} \colon \sum_{t=0}^{d-1} |a_t| \leq k \},
\]

where \( d \) is the dimension of the lattice, \( k \) is the template index, and \( |a_t| \) is the absolute value of the \( t \)th component of \( \vec{a} \). Figure 1(a) shows \( H_1^{(2)} \), the von Neumann template. The hatched cell is the template origin.

The second template of interest is defined as follows:

\[
J_k^{(d)} = \{ \vec{a} \colon -k \leq a_t \leq k \},
\]

Figure 1(b) shows \( J_1^{(2)} \), the Moore template.

The third template consists of the set of cells in one "corner" of the \( J_k^{(d)} \) template. This is

\[
L_k^{(d)} = \{ \vec{a} \colon 0 \leq a_t \leq k \},
\]

Figure 1(c) shows \( L_1^{(2)} \).

The fourth template consists of the outermost corner cells of \( J_k^{(d)} \) template. It is expressed as follows:

\[
M_k^{(d)} = \{ \vec{a} \colon a_t = \pm k \} \cup \{ \vec{0} \}.
\]

Figure 1(d) shows \( M_1^{(2)} \).
It is useful to describe two operations between templates and sets of points in general. Let $A$ and $B$ be two sets of points in $\mathbb{Z}^d$. Then, the sum of $A$ and $B$, $A + B$, is defined as follows:

$$A + B = \{ \vec{a} + \vec{b}; \; \vec{a} \in A, \; \vec{b} \in B \},$$

where $\vec{a} + \vec{b}$ is the $d$-tuple whose components are the respective sum of the components of $\vec{a}$ and $\vec{b}$.

No confusion should result between the three uses of the $+$ symbol, since context determines whether the $+$ operates on integers ($k, 1, 2, \text{etc.}$) as in integer addition or on points ($\vec{a}, \vec{b}, \text{etc.}$) or sets of points ($A, B, H_k^{(d)}$, etc.) As an example of the use of the sum operation between sets of points, for $A = B = f_{1}^{(d)}$, we have

$$f_{1}^{(d)} + f_{1}^{(d)} = f_{2}^{(d)}.
\tag{1}$$

Another useful operator is the multiply operator defined recursively as follows:

$$0A = \emptyset,
\quad (k + 1)A = kA + A,
\tag{2}$$

where $A$ is a set of points in $\mathbb{Z}^d$.

As an example, from (1) and (2), it follows that

$$2f_{1}^{(d)} = f_{2}^{(d)}.$$

Let a configuration $c(t)$ in a cellular space $Z$ denote an assignment of states at time $t$ to every machine in $Z$. Then, the configuration $c(t + 1)$ at the next time interval is determined uniquely from $c(t)$, $G$, and $N$. The configuration at $t = 0$, $c(0)$, is called the initial configuration. The configurations of interest here are those in which all but finitely many of the cells are in the quiescent state $q_0$. Let $C$ denote the set of all such configurations. Since the next state mappings of the machines in the array are restricted such that $f(q_0; q_0, q_0, \ldots, q_0) = q_0$, if $c(t) \in C$, it follows that $c(t + 1) \in C$. Let $F$ denote the function which maps each present configuration, $c(t)$, into the next configuration, $c(t + 1)$; i.e., $F(c(t)) = c(t + 1)$. $F$ will be called the global transition function. Further, the function $F'$ can be defined recursively, as follows:

$$F'(c(t)) = F(c(t)),
\quad F'(c(t)) = F(F'^{-1}(c(t))).$$
It is now possible to define the simulation of one cellular space by another. Let $\Phi_d$ be the set of global transition functions for $d - D$ spaces. Consider two spaces, $Z_1$ and $Z_2$, with configuration sets $C_1$ and $C_2$ and global transition functions $F_1$ and $F_2$, respectively. Then $S_2$ simulates $S_1$ if $k_2/k_1$ times real time if and only if there exists an effectively computable injective mapping $G: C_1 \rightarrow C_2$ and an effectively computable function $g: \Phi_d \rightarrow \Phi_d$ such that

$$F_2^{k_2}(G(e(t))) = G(F_1^{k_1}(e(t))),$$

where $F_2 = g(F_1)$.

The simulation is said to occur in real time, if $k_2 = k_1$.

For the next section of this paper, it is convenient to denote a $2 - D$ cellular space $Z$ by the pair $(T, r)$, where $T$ is the template and $r$ is the number of states per cell.

3. A SIMULATION OF A $(J^{(2)}_{m}, r)$ CELLULAR SPACE BY AN $(H^{(2)}_{k_m}, s)$ CELLULAR ARRAY SPACE IN $1/k$ TIMES REAL TIME

Before presenting the main result of this section, it is convenient to demonstrate the equivalence between certain templates. Lemma 1 below shows that the sum operation on two $J$ templates produces a single $J$ template.

**Lemma 1.** $J^{(d)}_m + J^{(d)}_n = J^{(d)}_{m+n}$ for $m \geq 1$ and $n \geq 1$.

**Proof.** Step (a). We show first $J^{(d)}_{m+n} \subseteq J^{(d)}_m + J^{(d)}_n$. Assume, without loss of generality, $m \geq n$, and let $\bar{a} \in J^{(d)}_{m+n}$, where $\bar{a} = (a_0, a_1, ..., a_{d-1})$. By definition of $J^{(d)}_{m+n}$, $-(m+n) \leq a_i \leq (m+n)$.

Form a vector $\bar{b} = (b_0, b_1, ..., b_{d-1})$ such that $b_1 = n$ if $0 \leq a_i \leq m+n$ and $b_1 = -n$ if $-(m+n) \leq a_i < 0$. Thus, $\bar{b} \in J^{(d)}_n$. Now form $\bar{c} = (c_0, c_1, ..., c_{d-1})$, such that $c_i = a_i - b_i$. There are two cases:

(i) $0 \leq a_i \leq m + n$, then $b_i = n$ and $-n \leq c_i \leq m$;

(ii) $-(m + n) \leq a_i < 0$, then $b_i = -n$ and $-m \leq c_i < n$.

But it is assumed that $m \geq n$, and so in either case (i) or (ii) $-m \leq c_i \leq m$. This is just the requirement for membership in $J^{(d)}_m$. Therefore, $\bar{c} \in J^{(d)}_m$. We have $\bar{a} = \bar{b} + \bar{c}$, where $\bar{b} \in J^{(d)}_n$ and $\bar{c} \in J^{(d)}_n$. Therefore, $\bar{a} \in J^{(d)}_m + J^{(d)}_n$ and $J^{(d)}_{m+n} \subseteq J^{(d)}_m + J^{(d)}_n$.

Step (b). Now demonstrate $J^{(d)}_m + J^{(d)}_n \subseteq J^{(d)}_{m+n}$. Let $\bar{f} \in J^{(d)}_m + J^{(d)}_n$, where $\bar{f} = (f_0, f_1, ..., f_{d-1})$. By definition of $J^{(d)}_m + J^{(d)}_n$, we have $\bar{f} = \bar{g} + \bar{h}$,
where \( \vec{g} = (g_0, g_1, \ldots, g_{d-1}) \), \( \vec{h} = (h_0, h_1, \ldots, h_{d-1}) \), \( \vec{b} \in J_n^{(d)} \), and \( \vec{h} \in J_m^{(d)} \). The components of \( \vec{f} \) can be expressed as follows: \( f_i = g_i + h_i \), where \( -m \leq g_i \leq m \) and \( -n \leq h_i \leq n \). But this implies \(-m+n \leq f_i \leq m+n\) and so, \( f \in J_{m+n}^{(d)} \). Thus, \( J_{m+n}^{(d)} \subseteq J_m^{(d)} \), and the lemma statement follows.

Q.E.D.

With respect to the multiply operator, the following statement is proved.

**Lemma 2.** \( kf_n^{(d)} = j_{nk}^{(d)} \) for \( k \geq 1 \) and \( n \geq 1 \).

**Proof.** The proof proceeds by induction on \( k \). By applying the definition of the multiply operation, it is seen that \( I f_n^{(d)} = f_n^{(d)} \). Now assume,

\[
k f_n^{(d)} = j_{nk}^{(d)}.
\]

By the definition of multiply operator

\[
(k+1) f_n^{(d)} = kf_n^{(d)} + j_n^{(d)}.
\]

Substituting (3) into (4) and applying Lemma 1 yields

\[
(k+1) f_n^{(d)} = j_{nk}^{(d)} + j_n^{(d)} = j_{(k+1)n}^{(d)}.
\]

Q.E.D.

Lemma 3 below shows the sum of the \( J_n^{(d)} \) and \( L_n^{(d)} \) template is equivalent to the sum of the \( M_n^{(d)} \) and \( L_n^{(d)} \) for a certain restriction on \( n \).

**Lemma 3.** \( J_n^{(d)} + L_n^{(d)} = M_n^{(d)} + L_n^{(d)} \) for \( m \geq 1 \) and \( n \geq 2m - 1 \).

**Proof.** **Step (a).** Since \( M_n^{(d)} \subseteq J_n^{(d)} \), \( M_n^{(d)} + L_n^{(d)} \subseteq J_n^{(d)} + L_n^{(d)} \).

**Step (b).** It is shown now that \( J_n^{(d)} + L_n^{(d)} \subseteq M_n^{(d)} + L_n^{(d)} \). Consider \( \vec{a} \in J_n^{(d)} + L_n^{(d)} \). Thus, \( \vec{a} = \vec{b} + \vec{c} \), where \( -m \leq b_i \leq m \) and \( 0 \leq c_i \leq n \). Then,

\[
-m \leq a_i \leq m + n.
\]

Form a vector \( \vec{b}' = (b_0', b_1', \ldots, b_{d-1}') \) such that \( b_i' = m \) if \( m \leq a_i \leq m + n \). Otherwise, \( b_i' = -m \). Form \( \vec{c}' = (c_0', c_1', \ldots, c_{d-1}') \) such that \( c_i' = a_i - b_i' \).

There are two cases:

(i) \( m \leq a_i \leq m + n \), then \( b_i' = m \) and \( 0 \leq c_i' \leq n \);

(ii) \( -m \leq a_i \leq m - 1 \), then \( b_i' = -m \) and \( 0 \leq c_i' \leq 2m - 1 \).

Since \( \vec{a} = \vec{b} + \vec{c} \) and \( \vec{b}' = \vec{b} + \vec{c}' \), then \( \vec{a} = \vec{b} + \vec{c} \) and the result is proved.

Q.E.D.
In the proof of the existence of a simulation, it is convenient to make use of the sufficiency condition adapted from Cole (1969) by Smith (1971). According to Lemma 3.5 of Smith (1971), a sufficient condition for the existence of a simulation of cellular space \( Z_t \) with template \( T \) by a cellular space \( Z_s \) with template \( T' \) in \( 1/k \) times real time is

\[
h(T') + K \supseteq hT + K,
\]

where \( K \subseteq \mathbb{Z}^d \) is a finite set of points and \( h \) is an injective homomorphism, \( h: \mathbb{Z}^d \to \mathbb{Z}^d \). The state set in \( Z_s \) at point \( \hat{p} \) is the Cartesian product of the state sets of the cells at \( (h(\hat{p}))+K \) in the simulated array.

Therefore, in order to determine the existence of a simulation in \( 1/k \) times real time, it is sufficient to find a set of points \( K \) and an injective homomorphism satisfying (5).

The \( h \) considered in this paper is

\[
h(\hat{x}') = m(x_1' + x_1', x_2' - x_1') = (x_1', x_2),
\]

where \( m \geq 1 \) and the prime indicates a point in \( Z_s \). \( h(\hat{x}') \) is seen to be an injective homomorphism as follows. First, \( h(\hat{x}') \) is an into mapping, since each point, \( \hat{x}' \), in \( Z_s \) has an image in \( Z_t \). Furthermore, each point in the simulated array can be seen to have a unique image in the simulating array as follows. Suppose, on the contrary, there exists at least two distinct points \( \hat{a} = (a_1, a_2) \) and \( \hat{b} = (b_1, b_2) \) such that \( h(\hat{a}) = h(\hat{b}) \). Then

\[
a_2 + a_1 = b_2 + b_1, \tag{7}
\]
\[
a_2 - a_1 = b_2 - b_1. \tag{8}
\]

Adding (7) and (8) yields

\[
a_2 = b_2. \tag{9}
\]

Subtracting (8) from (7) yields

\[
a_1 = b_1. \tag{10}
\]

But (9) and (10), of course, imply \( \hat{a} = \hat{b} \), contradicting the assumption that \( \hat{a} \) and \( \hat{b} \) are distinct. Thus, (6) is also a 1:1 mapping. Furthermore,

\[
h(\hat{a}' + \hat{b}') = h(a_1' + b_1', a_2' + b_2')
\]
\[
= m(a_2' + a_1' + b_2' + b_1', a_2' + b_2' - a_1' - b_1')
\]
\[
= m(a_2' + a_1', a_2' - a_1') + m(b_2' + b_1', b_2' - b_1')
\]
\[
= h(\hat{a}') + h(\hat{b}'),
\]
and
\[ h(ka') = h(ka_1', ka_2') = mh(a_1' + a_1', a_2' - a_1') = hh(a'). \]

Thus, \( h(x') \) is an injective homomorphism.

**Theorem 1.** For an arbitrary \( f(s, r) \) cellular space \( Z_1 \), there exists a \( (H_{2cm}^{(2)}, r^{(2cm)^2}) \) cellular space \( Z_2 \) which simulates \( Z_1 \) in \( 1/k \) times real time.

**Proof.** From Lemma 3.5 of Smith (1971), it is sufficient to show
\[ h(T') + K \supseteq kT + K \]
for \( T = f_m^{(2)} \) and \( T' = H_{2cm}^{(2)} \). Let \( K = L_{2cm-1}^{(2)} \) and choose \( h \) as in (6). The right side of (5) becomes
\[ kT + K = k_j^{(2)} + L_{2cm-1}^{(2)}. \]

Applying Lemma 2 to (11) yields,
\[ kT + K = f_m^{(2)} + L_{2cm-1}^{(2)}. \]

Now consider the left-hand side of (5). Template \( T' = H_{2cm}^{(2)} \) contains the five points \( (km, 0) \), \( (0, km) \), \( (-km, 0) \), \( (0, -km) \), and \( (0, 0) \). Under \( h \) these points map to \( (km, km) \), \( (km, -km) \), \( (-km, km) \), \( (-km, -km) \), and \( (0, 0) \), respectively, which is precisely \( M_{2cm}^{(2)} \). Thus,
\[ h(H_{2cm}^{(2)}) \supseteq M_{2cm}^{(2)}, \]
and so
\[ h(T') + K = h(H_{2cm}^{(2)}) + L_{2cm-1}^{(2)} \supseteq M_{2cm}^{(2)} + L_{2cm-1}^{(2)}. \]

Applying Lemma 3 to (12) yields,
\[ h(T') + K \supseteq f_m^{(2)} + L_{2cm-1}^{(2)}. \]

Thus, the sufficiency condition is satisfied.

The state set of each machine in the simulating space \( Z_2 \) is the Cartesian product of the states of \( K = L_{2cm-1}^{(2)} \). There are \((2km)^2\) machines in \( L_{2cm-1}^{(2)} \), each with \( r \) states. Thus, each machine in \( Z_2 \) has \( r^{(2cm)^2} \) states. Q.E.D.
Using a homomorphism of the form \( h(\vec{x}) = m \cdot \vec{x} \), Cole (1969) demonstrated a simulation of a \((J_1 d, r)\) space by an \((H_1 d, s)\) space in real time, where \( s = r^{216} \) for \( d = 2 \). Smith (1971), using a similar homomorphism, showed an improved simulation in which \( s = r^{48} \), for \( d = 2 \). Hamacher (1971) has improved this result showing a simulation in which \( s = r^{12} \). For the case of \( k = 1 \), Theorem 1 demonstrates a simulation of a \((J_1^{(2)}, r)\) space by an \((H_1^{(2)}, s)\) space, where \( s = r^4 \), a reduction by a factor of \( r^8 \) of the state set in the simulating space. Also, it is interesting to compare this with a result by Amoroso and Guilfoyle (1972), in which it is shown that a \((J_1^{(2)}, r)\) space can be simulated by a space with a neighborhood of just three cells in which the number of states per cell \( s \) is \( r^{45} \).

The result of this paper does not apply to general \( d - D \) spaces. For example, consider \( d = 3 \), \( m = 1 \), and \( k = 1 \). For these values,

\[
| H_1^{(d)} | = 7, \quad | L_{22n-1}^{(d)} | = 8,
\]

and so

\[
| h(H_1^{(d)}) + L_{22n-1}^{(d)} | \leq 56.
\]

However, \( | k J_n^{(3)} + L_{22n-1}^{(d)} | = 64 \). Thus, \( h(H_1^{(d)}) + L_{22n-1}^{(d)} \supset k J_n^{(d)} + L_{22n-1}^{(d)} \), and the sufficiency condition cannot be satisfied.

The prospect of a more optimum simulation, for \( d = 2 \), than the one presented here is an open question. It is interesting to note that for \( d = 2 \) there exists no \( K \) such that \( | K | \leq 2 \), which satisfies (5). For \( | K | = 2 \), \( | h(H_1^{(2)}) + K | \leq 10 \) and \( | J_1^{(2)} + K | \geq 12 \). Thus, \( h(H_1^{(2)}) + K \supset J_1^{(2)} + K \). Also, the sufficiency condition is not satisfied for \( | K | = 1 \). For \( | K | = 3 \), set order restrictions do not preclude satisfaction of the sufficiency condition.

4. Summary and Conclusions

The main result of this paper is a 2-dimensional simulation of a cellular space \( Z_1 \) with the \( J_1^{(2)} \) template by a space \( Z_2 \) with the \( H_2^{(2)} \) template in \( 1/k \) times real time. The simulation is accomplished by a relatively modest increase in cell complexity. Theorem 1 of this paper augments a result in Smith (1971), which proves the existence of a simulation of an arbitrary space \((T, r)\) by an \((H_1, S)\) cellular space in \( 1/k \) times real time for 2-D spaces. This result is based on two proofs:

(i) the existence of a simulation of a \((T, r)\) cellular space by a \((J_1, s')\) space in \( 1/k \) times real time, and

(ii) the existence of a simulation of a \((J_1, s')\) cellular space by an \((H_1, s)\) space in real time.
The cell complexity of the simulation of an arbitrary \((T,r)\) space by a \((H_k,s)\) space depends, of course, on the individual complexities of the two intermediate simulations. For the case \(d = 2\), Theorem 1 of this paper provides a more economical simulation for (2) above and, thus, for the general simulation

Received: September 15, 1973

References


