*Complex Systems* **6** (1992) 163–178

# Regular Language Invariance under One-Dimensional Cellular Automaton Rules

**Lenore Levine**
*Department of Mathematics, University of Illinois,*
*273 Altgeld Hall, 1409 W. Green St., Urbana, IL 61801, USA*

**Abstract.** Procedures are given for determining regular language invariance under one-dimensional cellular automaton rules. A metric is defined for the space of all one-dimensional cellular automaton rules over a given alphabet $\Sigma$. It is shown that under this metric, for certain regular languages, the set of rules under which the language is invariant contains no interior, and its complement contains no interior. Characteristics of surjective rules (rules under which the regular language $\Sigma^*$ is invariant) are also explored. Examples are given of a sequence of rules for which the limit language of the limit rule is not invariant under any rule in the sequence. Numerical experiments indicate that these rules do indeed display discontinuous behavior.

## 1. Introduction

Let $\Sigma^Z$ be the set of all mappings from the integers to some finite alphabet $\Sigma$ of $k$ characters; that is, the set of all doubly infinite sequences with entries in $\Sigma$. Then a cellular automaton is a function $f : \Sigma^Z \to \Sigma^Z$, such that if $y = f(x)$,

$$y_i = R(x_{i-r}, x_{i-r-1}, \ldots, x_{i+r}) \tag{1}$$

for some fixed function $R$ of $2r + 1$ variables. $R$, a *cellular automaton rule,* can be specified by a table of $k^{2r+1}$ entries; $r$ is said to be the *radius* of $R$, and site values are restricted to the alphabet $\Sigma = \{0, \ldots, k-1\}$.

Let $r_0$ be the minimal radius for which such a cellular automaton rule can be constructed for $f$. Then, for each radius $r \geq r_0$, one cellular automaton rule $R_r$ can be devised to simulate $f$, under the principle

$$R_r(x_{i-r}, x_{i-r-1}, \ldots, x_{i+r}) = R_0(x_{i-r_0}, x_{i-r_0-1}, \ldots, x_{i+r_0}); \tag{2}$$

that is, the first and last $r - r_0$ variables do not affect the value of the function.

The operation of cellular automaton rules can be generalized as follows. If $w = \alpha_1 \cdots \alpha_n$, $n \geq 2r+1$, then

$$R(w) = R(\alpha_1, \cdots, \alpha_{2r+1}) R(\alpha_2, \cdots, \alpha_{2r+2}) \cdots R(\alpha_{n-2r}, \cdots, \alpha_n). \tag{3}$$

Also, if $L$ is a subset of $\Sigma^*$,

$$R(L) = \{x : x = R(y), y \in L, |y| \geq r\} \tag{4}$$

and

$$R^{-1}(L) = \{x : y = R(x), y \in L\}. \tag{5}$$

This paper characterizes rules under which some regular language $L$ is invariant; that is, rules $R$ such that $R(L) = L$. (See [3] for definitions and characteristics of regular languages.) Section 2 discusses algorithms for proving or disproving invariance of $L$ under $R$.

Observations have shown that the behavior of rules may change drastically if only a few entries in the rule table are changed. In order to address such abrupt changes in qualitative behavior, we are led to define a metric on the space of one-dimensional cellular automaton rules over a given alphabet. This metric is defined in section 3. We use this metric to characterize the set of cellular automaton rules under which a given regular language is invariant.

In section 4 a possible relevance of regular language invariance to long-term behavior of cellular automaton rules is explored. A rule $R$ is presented such that when $R$ is repeatedly applied to an infinite lattice, every finite section of that lattice clearly evolves to a member of a regular language $L$. It is demonstrated that there are members of cellular automaton rule space arbitrarily close to $R$, on a given path, for which $R(L) \neq L$. By numerical experiment, it is shown that rules extremely close to $R$ on this path have long-term behavior very different from $R$. Another sequence of rules is presented that converges to the limit rule under the metric presented in section 3, but that can be shown not to converge in behavior.

## 2. Invariance algorithms

Wolfram [5] describes an algorithm for finding a finite automaton characteristic of $R(\Sigma^*)$. A Mealy machine (a finite automaton that produces output on state transition; for more information see [3]) is constructed with $k^{2r}$ states, one state for each string of length $2r$ in $\Sigma^*$. Any of these states can be the start state. Arrows lead out of these states on each character in $\Sigma$ as follows: Let $w = \alpha_1 \cdots \alpha_{2r}$ have length $2r$; let $S(w)$ be the corresponding state. Arrows lead out of $S(w)$ on each member of $\Sigma$ as follows: If $R(w\beta) = \gamma$, then there is an arrow leading out of $S(w)$ with label $\beta$ to $S(\alpha_2 \cdots \alpha_{2r}\beta)$. The state transition output is $\gamma$. This nondeterministic "transducer," or Mealy machine, can be converted by standard constructions to a minimal deterministic finite automaton (DFA) (see [3], chapter 2).

This procedure can be generalized to create a Mealy machine $M$ that, for any regular language $L$, outputs $R(L)$. Let $D$ be the DFA that recognizes $L$.

It is possible, by following arrows backward, to determine, for any state $T$ of $D$, which sequences of $2r$ characters are the last $2r$ characters of a string with final state $T$. Call such sequences $F(T)$. For each combination of state $T$ of $D$ and sequence $w = \alpha_1 \cdots \alpha_{2r}$ in $F(T)$, let a state $S(T, w)$ be part of $M$. If there is an arrow leading out, in $D$, from state $T$ to state $U$, on character $\beta$, and $R(w\beta) = \gamma$, then let there be an arrow leading out on character $\beta$ from state $S(T, w)$ to state $S(U, \alpha_2 \cdots \alpha_{2r}\beta)$. Let $\gamma$ be output on this transition. Let $S(T, w)$ be accepting if $T$ is accepting. Finally, if $T$ is the start state of $D$, for any $w$ such that $S(T, w)$ exists, let $S(T, w)$ be a start state of $M$.

It can be shown by induction that $M$ will output any nonempty string $x$ if and only if $x$ is in $R(L)$. Standard constructions in [3] can be used to convert $M$ to a minimal DFA. Since a minimal DFA is unique [3], it is possible to determine whether $R(L) = L$. Using algorithms to find the union, intersection, and complements of regular languages ([3], chapter 3), it can also be determined if $R(L) \subseteq L$ or $L \subseteq R(L)$. (Another presentation of this algorithm is found in [1].)

If $L$ is a regular language, it is also possible to determine invariance of $L$ under $R$ without explicitly finding $R(L)$, by testing all strings of a given length or less in $L$. This length is dependent only on the radius of $R$, the number of characters in the alphabet of $L$, and the number of states in the DFA of $L$.

**Theorem 1.** *Let $L_1$ and $L_2$ be regular languages using an alphabet $\Sigma$ of $k$ characters. Let them be accepted, respectively, by DFAs $D_1$ and $D_2$ with $n_1$ and $n_2$ accepting states. Let $R$ be a $k$-symbol, $r$-radius one-dimensional cellular automaton rule. Let $q = n_1 n_2 k^{2r} + 2r$, and let $A(L_1, q+1)$ be all strings of length $q+1$ or less in $L_1$. Then if $R(A(L_1, q+1)) \subseteq L_2$, $R(L_1) \subseteq L_2$.*

**Proof.** Assume we have shown that $R(A(L_1, m-1)) \subseteq L_2$ for $m - 1 > q$. We wish to show $R(A(L_1, m)) \subseteq L_2$.

Let $w$ be any string in $L_1$ such that $2r + 1 \leq |w| \leq m - 1$. Then the sequence-state triple associated with $w$ consists of (a) the last $2r$ characters of $w$, (b) the state $D_1$ is in when it accepts $w$, and (c) the state $D_2$ is in when it accepts $R(w)$.

Let $|w| = m - 1$. Since (a) $m - 1 > q$, (b) there are only $q - 2r$ possible sequence-state triples, and (c) every string of length $2r + 1$ or greater has a sequence-state triple associated with it, $\exists$ some $x$ such that $x$ is a prefix of $w$, and $x$ has the same state-sequence triple associated with it as $w$.

Let $\gamma_1, \ldots, \gamma_j$ be those members of $\Sigma$ such that $x\gamma_i, 1 \leq i \leq j$, is in $L_1$. Then these are the only characters that when added to $w$ produce a string in $L_1$, since $w$ ends in the same state as $x$.

Let $\beta_i = R$(last $2r$ characters of $x$, followed by $\gamma_i$). Then for each $\beta_i$, since $|x\gamma_i| \leq m - 1$, $R(x)\beta_i$ is in $L_2$. Therefore, from the state $D_2$ is in when it accepts $R(x)$, there is a path out on each $\beta_i$. Since $w$ has the same last $2r$ characters as $x$, $\beta_i$ is also $R$(last $2r$ characters of $w$, followed by $\gamma_i$). And since $D_2$ is in the same state when it accepts $R(w)$ as when it accepts

$R(x), R(w)\beta_i$ is in $L_2$ for each $\beta_i$, and hence $R(w\gamma_i)$ is in $L_2$ for all possible $\gamma_i$ that can follow $w$. Since this is true for all $w$ such that $|w| = m$, $w$ in $L_1$, $R(A(L_1, m) \subseteq L_2$. ∎

**Corollary.** *If* $R(A(L_1, (n_1)^2k^{2r} + 2r + 1)) \subseteq L_1$, $R(L_1) \subseteq L_1$.

**Corollary.** *Let* $R^j$ *be the* $j$*th iterate of* $R$. *If* $R^j(A(L_1, n_1n_2k^{2rj} + 2rj + 1)) \subseteq L_2$, $R(L_1) \subseteq L_2$.

**Proof.** $R^j$ can be regarded as a cellular automaton rule of radius $rj$. ∎

Note that $|A(L_1, q+1)|$ is an upper bound of the number of strings that need to be tested to show $R(L_1) \subseteq L_2$, but it is not a least upper bound. As a matter of fact, if $L_1 \neq \Sigma^*$, this number can be shown *not* to be a least upper bound.

**Theorem 2.** *If* $L_1 \neq \Sigma^*$ *and* $R(L_1) \not\subseteq L_2$, *it is never necessary to test all strings in* $A(L_1, q+1)$ *to show* $R(L_1) \not\subseteq L_2$.

**Proof.** Since $L_1 \neq \Sigma^*$, there are some states that have fewer than $k$ arrows going out to accepting states.

Now, in order for the above algorithm to fail first on sequences of length $n_1n_2k^{2r} + 2r + 1$, some string of length $n_1n_2k^{2r} + 2r$ must contain every possible combination of accepting state and last $2r$ characters. That is, every accepting state must have $k$ arrows leading into it. But this is not possible, since some of the accepting states have fewer than $k$ arrows leading to other accepting states. ∎

**Theorem 3.** *Let* $L_1$, $L_2$, $D_1$, $D_2$, $n_1$, *and* $n_2$ *be as above. Let* $R$ *be a* $k$*-state,* $r$*-radius one-dimensional cellular automaton rule. Let* $q = n_2 2^{n_1 k^{2r}}$, *and let* $A(L_2, q+1)$ *be all strings of length* $q+1$ *or less in* $L_2$. *Then if* $A(L_2, q+1) \subseteq R(L_1)$, $L_2 \subseteq R(L_1)$.

**Proof.** Assume we have shown that $A(L_2, m-1) \subseteq R(L_1)$ for $m - 1 > q$. We wish to show $A(L_2, m) \subseteq R(L_1)$.

Let $w$ be any string in $L_2$ such that $|w| \leq m - 1$. Let the state-sequence pairs associated with $w$ consist of, for each $x$ in $R^{-1}(w)$, (a) the last $2r$ characters of $x$ and (b) the state $D_1$ is in when it accepts $x$. Let $S(w)$ be the state $D_2$ is in when it accepts $w$.

Now, let $|w| = m - 1 > q = n_2 2^{n_1 k^{2r}}$. Since there are only $n_2$ possibilities for $S(w)$, and only $2^{n_1 k^{2r}}$ possibilities for the state-sequence pairs associated with $w$, there must be some $y$ such that $y$ is a prefix of $w$, $S(y) = S(w)$, and $y$ has the same state-sequence pairs associated with it as $w$.

We know that the final state of $y$ is the same as that of $w$. Therefore $w\gamma$ is in $L_2$ if and only if $y\gamma$ is in $L_2$. Let $w\gamma$ be in $L_2$. Since $y\gamma$ is in $L_2$ and $|y\gamma| \leq m - 1$, it has a preimage in $L_1$; specifically, it is $R(x\beta)$ for some $x$ in $R^{-1}(y) \cap L_1, x\beta$ in $L_1$. This means that the final state of $x$ is such as to allow appendage of a character $\beta$; and it also means that $R$(last $2r$ characters of $x$, followed by $\beta$) $= \gamma$.

Also, because $y$ and $w$ have the same state-sequence pairs associated with them, we know that there is some string $s$ in $R^{-1}(w)$ that has the same last $2r$ characters as $x$ and ends in the same state as $x$. Since $s$ ends in the same state as $x$, $s\beta$ is in $L_1$; since it has the same last $2r$ characters as $x$, $R$(last $2r$ characters of $s$, followed by $\beta$) $= \gamma$, and $R(s\beta) = w\gamma$. Hence $w\gamma$, for any possible $\gamma$ that can follow $w$ in $L_2$, has a preimage in $L_1$. Since $w$ is an arbitrary string of length $m - 1$ in $L_2$, all strings of length $m + 1$ in $L$ have a preimage in $L$. ∎

**Corollary.** *If* $A(L_1, n_1 2^{n_1 k^{2r}} + 1) \subseteq R(L_1)$, $L_1 \subseteq R(L_1)$.

**Corollary.** *If* $A(L_2, n_2 2^{n_1 k^{2rj}} + 1) \subseteq R^j(L_1)$, $L_2 \subseteq R^j(L_1)$.

## 3. Cellular automaton rule space

Any cellular automaton rule $R$ of radius $r$ can be considered equivalent to a rule of radius $r'$, $r' > r$, under the principle

$$R(\beta\alpha\gamma) = R(\alpha) \tag{6}$$

for all $\alpha$, $\beta$, and $\gamma$ in $\Sigma^*$ such that $|\alpha| = r$ and $|\beta|, |\gamma| = r' - r$. (Note that both $R$ and $R'$ represent the same cellular automaton function.)

Let a metric be defined on the space of $k$-state cellular automaton rules ($S_k$) as follows: Let $R_1$ and $R_2$ be $k$-state cellular automaton rules of radii $r_1$ and $r_2$, respectively, with $r_1 \leq r_2$. Let $R_1'$ be the rule equivalent to $R_1$ of radius $r_2$; and let $d(R_1, R_2)$ be the proportion of table entries in which $R_1'$ and $R_2$ differ. Note that $d$ is consistent if we consider rules of radius $r_3 > r_2$ that are equivalent to $R_1$ and $R_2$. The two rules will differ in $k^{2(r_3 - r_2)}$ times as many table entries, and there are $k^{2(r_3 - r_2)}$ times as many total table entries.

**Theorem 4.** *d is a metric.*

**Proof.** Let $R_1$, $R_2$, and $R_3$ be $k$-state cellular automaton rules. By the previous paragraph, $d$ can be calculated by regarding all three rules as having radius $r$, the maximum of their three radii. Let $S(R_i, R_j, r)$ be the amount of table entries differing between $R_i$ and $R_j$, if they are both regarded as having radius $r$. Note that

$$d(R_i, R_j) = S(R_i, R_j, r)/k^{2r+1}. \tag{7}$$

Any entry in which $R_1$ and $R_3$ differ, thus incrementing $S(R_1, R_3, r)$, must also increment either $S(R_1, R_2, r)$, $S(R_2, R_3, r)$, or both. Thus

$$S(R_1, R_3, r) \leq S(R_1, R_2, r) + S(R_2, R_3, r); \tag{8}$$

and hence

$$d(R_1, R_3) = d(R_1, R_2) + d(R_2, R_3). \;∎ \tag{9}$$

Let $L$ be a regular language; and let $I_L$ be the set of those $k$-symbol ($k \geq 2$) cellular automaton rules under which $L$ is invariant; that is, for which $R(L) = L$. It is possible to prove results about $I_L$ for certain regular languages.

The regular languages we are most concerned with are those that might characterize all finite subsequences of a doubly infinite sequence; and these languages have the characteristic that if some string $w$ is in such a language $L$, every proper substring of $w$ is also in $L$. They also have the characteristic that if $w$ is in $L$, there is a $\alpha_1 w \alpha_2$ in $L$ for some $\alpha_1$ and $\alpha_2$ in $\Sigma$; and therefore, there is a $\beta w \gamma$ in $L$, $|\beta| = n_1$, and $|\gamma| = n_2$ for any $n_1$, $n_2 \geq 0$. Let such languages be called *data-stream-like* languages. Note that if cellular automaton rules $R$ with radius $r$, and $R'$ with radius $r' > r$, represent the same cellular automaton function, and $L$ is a data-stream-like language, $R(L) = R'(L)$. If $|w| \geq 2r' + 1$, the operations of $R$ and $R'$ on $w$ are equivalent; and if $2r + 1 \leq |w| < 2r' + 1$, there is a $\beta w \gamma$ in $L$ such that $|\beta|$, $|\gamma| = r' - r$; and since $R$ and $R'$ are equivalent, $R'(\beta w \gamma) = R(w)$.

**Lemma 1.** *Let $L$ be a regular data-stream-like language that does not equal $\Sigma^*$. Then*

$$\lim_{n\to\infty} \frac{|\{w \in L : |w| = n\}|}{|\{w : |w| = n\}|} = 0. \tag{10}$$

**Proof.** Let $w$ be a word in $\bar{L}$. Since $L$ is data-stream-like, any member of $\{xwz : x, z \in \Sigma^*\}$ is in $\bar{L}$. Therefore,

$$\begin{aligned} &|\{y : y \in L, |y| = n\}| && (11)\\ &\quad \leq |\{y : y \neq xwz, |y| = n\}| && (12)\\ &\quad \leq |\{y : y \neq xwz, |x| \cong 0 \bmod |w|, |y| = n\}| && (13)\\ &\quad = (k^{|w|} - 1)^{\lfloor n/|w| \rfloor}. && (14)\end{aligned}$$

Therefore,

$$\begin{aligned} &\lim_{n\to\infty} \frac{|\{w \in L : |w| = n\}|}{|\{w : |w| = n\}|} && (15)\\ &\quad \leq \lim_{n\to\infty} \frac{(k^{|w|} - 1)^{\lfloor n/|w| \rfloor}}{k^n} && (16)\\ &\quad \leq \lim_{n\to\infty} \frac{(k^{|w|} - 1)^{n/|w|}}{k^n} = 0. \blacksquare && (17)\end{aligned}$$

**Theorem 5.** *If $L$ is data-stream-like, $I_L$ has no interior.*

**Proof.** Let $R$ be any rule in $I_L$, and let $R$ have radius $r$. Let $\epsilon$ be any number $> 0$. By the preceding lemma, there is an $r' \geq r$ such that

$$\frac{|\{w \in L : |w| = 2r' + 1\}|}{|\{w : |w| = 2r' + 1\}|} < \epsilon. \tag{18}$$

Let $R'$ be the rule equivalent to $R$ of radius $r'$.

Since $k \geq 2$, $\exists$ some symbol $\alpha$ such that $L \neq \alpha^*$. Let $R_2$ be a rule of radius $r'$ such that for $|y| = 2r' + 1$, $R_2(y) = \alpha$ if $y \in L$; and $R_2(y) = R'(y)$ if $y \notin L$. Then $d(R, R_2) < \epsilon$; and since $L$ is data-stream-like, and there are thus strings in $L$ of any length, $R_2(L) = \alpha^*$, and not $L$. ∎

**Theorem 6.** *If $L$ is data-stream-like, $\bar{I}_L$ has no interior.*

**Proof.** Let $R$ be a rule in $\bar{I}_L$ of minimal radius $r$. Let $\epsilon$ be any number $> 0$. By the preceding lemma, $\exists$ an $r' \geq r$ such that

$$\frac{|\{w \in L : |w| = 2r' + 1\}|}{|\{w : |w| = 2r' + 1\}|} < \epsilon. \tag{19}$$

Let $R'$ be the rule equivalent to $R$ of radius $r'$. Let $I_{r'}$ be the identity rule of radius $r'$; that is,

$$I_{r'}(\alpha_{-r'} \cdots \alpha_0 \cdots \alpha_{r'}) = \alpha_0. \tag{20}$$

Let $R_2$ be a rule of radius $r'$ such that for $|y| = 2r' + 1$, $R_2(y) = I_r(y)$ if $y \in L$; and $R_2(y) = R'(y)$ if $y \notin L$. Then $d(R, R_2) < \epsilon$; and $R_2(L) = L$. ∎

**Corollary.** *If $L$ is data-stream-like, $I_L$ ($\bar{I}_L$) contains no isolated points.*

**Proof.** If there were an isolated point in $I_L$ ($\bar{I}_L$), an annulus surrounding it would be in $\bar{I}_L$ ($I_L$). ∎

**Theorem 7.** *$I_{\Sigma^*}$ has no interior.*

**Proof.** Let $R$ be a $k$-state cellular automaton rule of radius $r$ in $I_{\Sigma^*}$; that is, every string $w$ in $\Sigma^*$ has a preimage under $R$. Let $R'$ be the equivalent rule of radius $r' > r$. $R'$ is also surjective; that is, in $I_{\Sigma^*}$. According to [2], this occurs only if all strings in $\Sigma^*$ have the same number of preimages under $R'$. Specifically, all characters $\alpha$ in $\Sigma$ must have $k^{2r'}$ preimages; that is, for each $\alpha$ in $\Sigma$, there must be $k^{2r'}$ entries in the rule table that go to $\alpha$. It is only necessary to change one of these entries to take the rule out of $I_{\Sigma^*}$; thus, there is a rule $R_2$ in $\bar{I}_{\Sigma^*}$ such that $d(R, R_2) = 1/r'$. Since $r'$ can be made arbitrarily large, $R$ is not in the interior of $I_{\Sigma^*}$. ∎

**Lemma 2.** *Let $R$ be a $k$-state cellular automaton rule of radius $r$. Changing any table entry for $R$ will change at most $f(k, n) = nk^{n-1}$ images of strings of length $n + 2r$; that is, $f(k, n)$ preimages of strings of length $n$.*

**Proof.** Changing one table entry means changing $R(z)$, $|z| = 2r + 1$. $z$ is in position $i$ in $k^{n-1}$ strings of length $n + 2r$, $1 \leq i \leq n$. Therefore $z$ can be in at most $nk^{n-1}$ strings of length $n + 2r$. ∎

**Corollary.** *To change $m$ preimages of strings of length $n$, it is necessary to change at least $\lceil m/f(k, n) \rceil$ table entries.*

**Theorem 8.** *$I_{\Sigma^*}$ is closed.*

**Proof.** It suffices to show that if a $k$-state rule $R$ is in $\bar{I}_{\Sigma^*}$, $\exists$ an $\epsilon > 0$ such that if $d(R, R_2) < \epsilon$, $R_2 \in \bar{I}_{\Sigma^*}$.

Let $R$ in $\bar{I}_{\Sigma^*}$ have radius $r$. By [2], there is some string $w$ in $\Sigma^*$ such that $w$ has $z$ excess preimages under $R$; that is,

$$|\{y : R(y) = w\}| - \frac{k^{|w|+2r}}{k^{|w|}} = z, \ z > 0. \tag{21}$$

To change $R$ to a surjective rule (rule in $I_{\Sigma^*}$) $R_2$ of radius $r$, it is necessary to change at least $z/f(k, |w|)$ table entries; so

$$d(R, R_2) \geq \frac{z}{f(k, |w|)k^{2r+1}}. \tag{22}$$

Now, let us consider the rule $R'$, of radius $r' > r$, equivalent to $R$. The rule table of $R'$ contains $zk^{2(r'-r)}$ excess preimages of $w$; so to change $R'$ to a surjective rule $R_2$ of radius $r'$, it is necessary to change at least $zk^{2(r'-r)}/f(k, |w|)$ table entries. Therefore,

$$d(R', R_2) = d(R, R_2) \geq \frac{zk^{2(r'-r)}}{f(k, |w|)k^{2r'+1}} = \frac{z}{f(k, |w|)k^{2r+1}}. \ \blacksquare \tag{23}$$

It is also possible to show that large numbers of points in $I_{\Sigma^*}$ are not isolated. In order to do so, it is necessary to define the effective diameter of a rule. A one-dimensional cellular automaton rule $R$ of radius $r$ has effective diameter $d$ if $R(a_1 w b_1) = R(a_2 w b_2)$ for all $a_1$, $a_2$, $w$, $b_1$, and $b_2$ such that $|a_1| = |a_2|$, $|b_1| = |b_2|$, $|w| = d$, and $|a_1| + |w| + |b_1| = 2r + 1$. Such a rule can, therefore, be considered equivalent to a function $R'$ from $d$ variables in $\Sigma$ to $\Sigma$. Both $R$ and $R'$ implement the same cellular automaton function $f$; that is, if $y \in \Sigma^Z = f(x)$,

$$y_i = R'(x_{i-r+|a_1|} \ldots x_{i+r-|b_1|}). \tag{24}$$

Let $R$ be a rule of effective diameter $d$ operating on an alphabet $\Sigma$ of $k$ letters, equivalent to a rule $R'$ that is a function from $d$ variables in $\Sigma$ to $\Sigma$. Then $R$ is left permutive if $R'(p\alpha_1), \ldots, R'(p\alpha_k)$ permute the elements of $\Sigma$, for all $p$ such that $|p| = d - 1$. Right permutive rules are defined similarly (see [4]).

**Theorem 9.** *If $R$ is a left or right permutive rule, it is not an isolated point in $I_{\Sigma^*}$; that is, there are surjective rules arbitrarily close to it.*

**Proof.** Let $n$ be any nonnegative integer, and let $p_0$ be a string of length $n$. Let $R$ be a left permutive rule of effective diameter $d$ over an alphabet $\Sigma$ of $k$ letters; and let $q_0$ be a string of length $d$. Let $R'$ be the equivalent rule accepting $d$ characters. Let $R_2$ be a rule of effective diameter $d+n$ defined as follows: $R_2(pq\alpha) = R'(q\alpha)$ for all $p$, $q$, and $\alpha$ such that $|p| = n$, $|q| = d$, and $|\alpha| = 1$, unless $p = p_0$ and $q = q_0$. In this case, let $R_2(p_0 q_0 \alpha_1), \ldots, R_2(p_0 q_0 \alpha_k)$

form a permutation of $\Sigma$ different from that of $R'(q_0\alpha_1), \ldots, R'(q_0\alpha_k)$. Then $R_2$ is also a left permutive rule, and hence surjective; and

$$d(R, R_2) = d(R', R_2) = \frac{k}{k^{d+n}}, \tag{25}$$

which can be made arbitrarily small by increasing $n$. ■

This theorem leads to the following conjecture.

**Conjecture 1.** *$I_{\Sigma^*}$ contains no isolated points.*

## 4. Rules with discontinuous behavior

### 4.1 The *GC* sequence

In the preceding sections, we showed that for any regular language $L$, and for any rule $R$ under which $L$ is invariant, there are rules arbitrarily close to $R$ under which $L$ is not invariant. This suggests that if $L$ is the limit language of a rule, or a very large part of such a limit language, changing an arbitrarily small proportion of table entries may completely change the behavior of the automaton. A case is presented in which rule behavior is, indeed, discontinuous.

One-dimensional, two-state, radius 1 cellular automata are specified by an eight-digit binary number in which the leftmost digit is $R(111)$, followed by $R(110), \ldots, R(000)$ (see [6]).

We can see, therefore, that $R_{128}(111) = 0$; and $R_{128}(w) = 0$ for $|w| = 3$ and $w \neq 111$. Thus, unless an infinite lattice initially contains all ones, any finite part of the lattice will, after enough generations, contain only members of the regular language $L = 0*$, which is invariant under this rule. This process is likely to happen quite quickly; for example, unless a finite part of the lattice contains at least 101 ones in a row, it will regress to all zeros in less than fifty generations.

The Grand Canyon (*GC*) sequence of rules (so named because of the rules' appearance) is constructed as follows: $GC_r$, $r \geq 2$, is exactly like rule 128 except $GC_r(0^{2r+1}) = 1$. Thus, this series converges to rule 128; but $L$ is not invariant under any member of this series.

Experimental work (see figures 1, 2, and 3) was done on a 640- or 1280-cell wide cross-section of a doubly infinite lattice. (To show the effects of a $r$-radius rule after $g$ generations, it is necessary to start out with $640+2rg$ cells.) This work shows that members of the *GC* series exhibit entirely different behavior from rule 128. Under random initial conditions (created by the rule 31 random number generator described in Appendix A) a significant number of ones continue to appear on the lattice after hundreds of generations. This is true even for $GC_{100}$, which has a distance from rule 128 of only $2^{-201}$.

Figure 4 shows the proportion of ones, in generations 11 through 400, when rules in the *GC* sequence are run with initial conditions as given above.

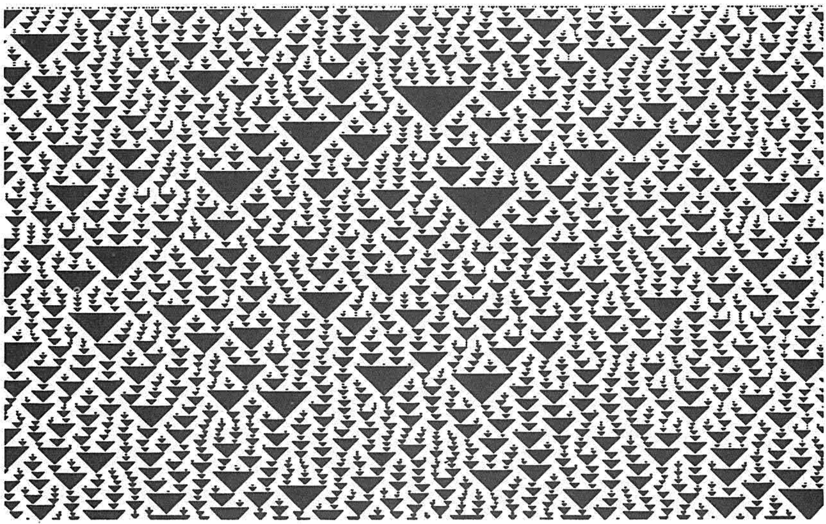

Figure 1: Rule $GC_7$. This rule differs from rule 128 only in that if a cell is zero-valued, as are its seven neighbors on either side, it becomes one-valued in the next generation. Its distance from rule 128 is $2^{-15}$.

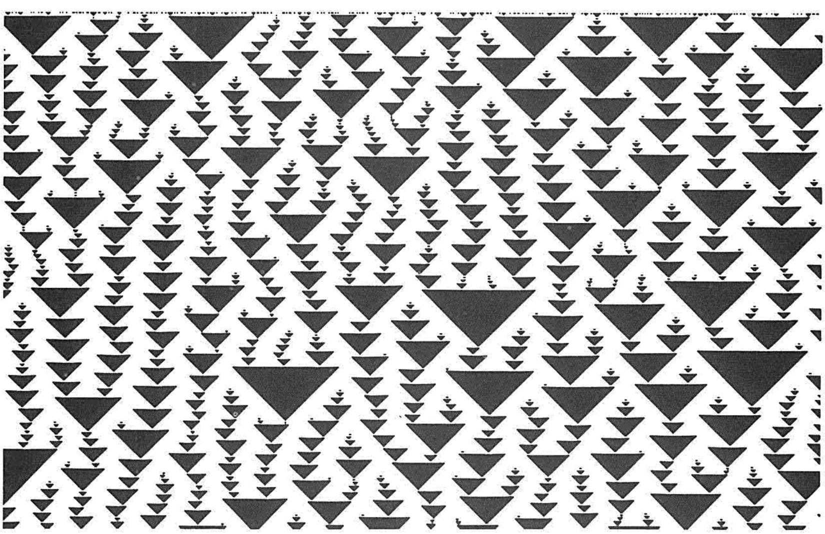

Figure 2: Rule $GC_{17}$. This rule differs from rule 128 only if a cell and its seventeen neighbors on either side are zero-valued. Its distance from rule 128 is $2^{-35}$.

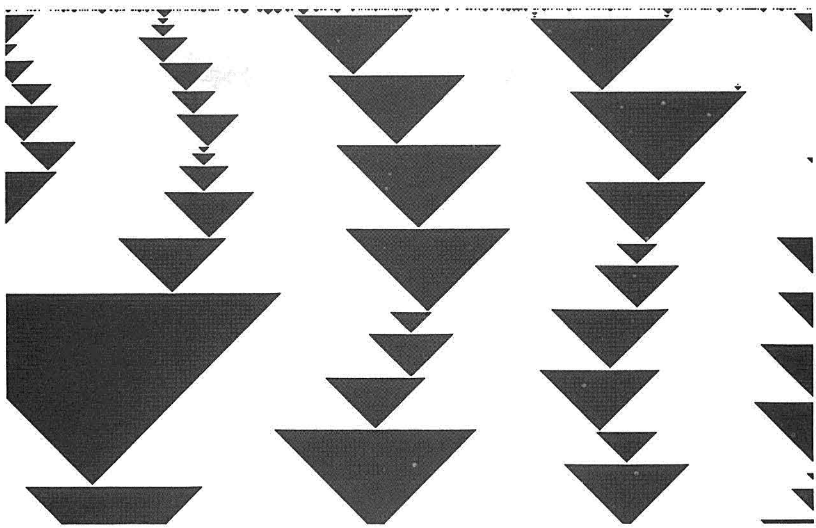

Figure 3: Rule $GC_{100}$. This rule differs from rule 128 only if a cell and its 100 neighbors on either side are zero-valued. Its distance from rule 128 is $2^{-201}$.

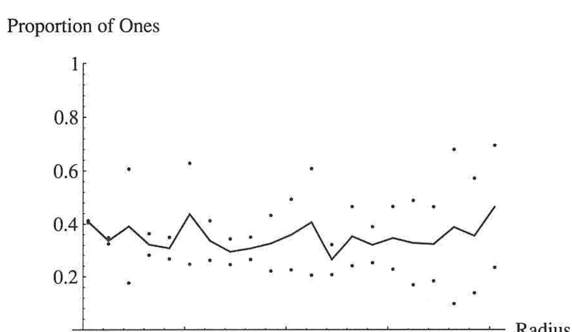


Figure 4: The proportion of ones in generations 11 through 400, when rules in the $GC$ series are run on a 1280-cell wide section of a doubly infinite lattice. Rules with radii congruent to 5 mod 20 are tested, from $GC_5$ to $GC_{405}$. The center line represents the average proportion of ones over ten runs, with outside dots representing the standard deviation. This proportion remains significant even under $GC_{405}$, which has a distance from Rule 128 of only $2^{-811}$.

It is possible, for any rule in this series $GC_k$, to select initial conditions $I_k$ such that as $k$ approaches infinity, the behavior of $GC_k$ on $I_k$ comes arbitrarily close to the behavior of rule 128. Simply select every $2k+2$th cell to be a one. In the next generation also, every $2k+2$th cell (those having $k$ zero-valued neighbors on each side in the first generation) will be a one, and the rest of the cells will be zero-valued.

However, it may be that these initial conditions are atypical. The above experimental evidence leads to the conjecture that for most initial conditions, a significant number of ones continue to appear in the lattice under all rules in this series. This is expressed more formally in the following.

**Conjecture 2.** *Let $Z(I, R, t)$ be the proportion of zeros on a doubly infinite lattice with initial conditions $I$, in generation $t$, under the application of cellular automaton rule $R$. Then, there is an $\alpha > 0$ and a $\beta < 1$ such that, for almost all initial conditions $I$, and for each rule in the GC sequence $GC_m$,*

$$\liminf_{t\to\infty} Z(I, GC_m, t) > \alpha \tag{26}$$

*and*

$$\limsup_{t\to\infty} Z(I, GC_m, t) < \beta. \tag{27}$$

Experimental evidence also suggests that, as the distance between rules in the $GC$ sequence and rule 128 goes to zero, use of the anomalous table entries does go to zero, even though in each generation, under each rule in the sequence, there continues to be a large proportion of cells showing the effect of anomalous table entry use in preceding generations. That is, they have previously been zero-valued, and now have a value of one. Figure 5 shows the proportion of times anomalous table entries were used, under the same initial conditions as in the previous experiment. These experiments lead to Conjecture 3.

**Conjecture 3.** *Let $A(I, R_1, R_2, t)$ be the proportion of times table entries differing from $R_1$ are used, when rule $R_2$ is applied to a doubly infinite lattice, under initial conditions $I$, in generation $t$. Then, for almost all initial conditions $I$,*

$$\lim_{t\to\infty, m\to\infty} A(I, \text{Rule}\,128, GC_m, t) = 0. \tag{28}$$

However, for conjecture 2 to be true,

$$\lim_{t\to\infty} A(I, \text{Rule}\,128, GC_m, t) \tag{29}$$

must be nonzero for each $m$.

Note that given two reasonable assumptions about the behavior of rules in this sequence, conjecture 3 must be true. The first assumption is conjecture 2. The second assumption comes from observation of figures 1, 2, and 3. Each figure consists of black triangles against a white background; and the

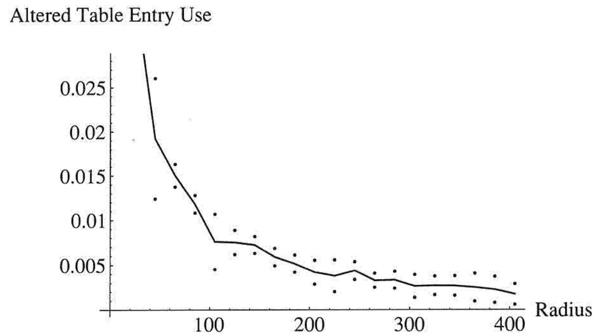


Figure 5: The proportion of times the table entry differing from rule 128 is used in generations 11 through 400, on a 1280-cell wide cross-section of a doubly infinite lattice. Rules with radii congruent to 5 mod 20 are tested, from $GC_5$ to $GC_{405}$. The center line represents the average proportion over ten runs, with outside dots representing the standard deviation.

average area of the triangles appears to be proportional to the radius of the rule. The second assumption is that this is indeed the case.

Black areas in the above figures indicate cells that are one-valued in a particular generation. There are two reasons a cell operated on by $GC_n$ can be one-valued in generation $k$: either because it was, in generation 0, one-valued and surrounded by at least $k$ one-valued cells on each side; or because, in some previous generation, it was zero-valued and surrounded by more than $n$ zero-valued cells on each side. As $k$ goes to infinity, the number of cells that are one-valued for the first reason, in generation $k$, goes to zero. Let us say in the second case that the black area is part of a nontransient triangle. Under any of the $GC$ rules, under almost all initial conditions, after a few generations all but a very small proportion of the one-valued cells will be part of nontransient triangles.

Now consider $n$ very large; by our second assumption, the area of the nontransient triangles will usually be quite large. Since the number of generations a triangle lasts equals the width $w$ of the top of the triangle, the area of each triangle should equal $w^2/2$. Therefore, if the areas of the triangles are usually quite large, the proportion of the width of their tops to their areas should usually be quite small. Now, each black cell at the top of a nontransient triangle indicates the use of an anomalous table entry; and anomalous table entries are used only in such cases. Therefore the proportion of times anomalous table entries are used (the proportion of times cells are black and at the top of a nontransient triangle) should be very small compared to the proportion of times cells are part of a nontransient triangle. And this is very close to the proportion of one-valued cells.

This experiment evokes some general questions about the actions of one-dimensional cellular automaton rules. That is, if a sequence of rules

$$R_1, R_2, \ldots, R_n, \ldots \tag{30}$$

is observed to behave discontinuously from the limit rule $R_0$, must it be because

$$\lim_{t \to \infty} A(I, R_0, R_m, t) \tag{31}$$

is nonzero for each $m$? That is, must anomalous table entries be part of the nontransient behavior of each rule? The next example shows that this is not, in fact, the case.

### 4.2 The *RB* sequence

The $RB$ sequence is a sequence of one-dimensional cellular automaton rules that converges, under the metric given earlier in the paper, to a limit rule; but for which each member behaves very differently from the limit rule.

The limit rule, $RB_\infty$, is a rule over an alphabet of four letters: (dark, red), (dark, blue), (light, red) and (light, blue). It is regarded as the product of two different two-letter rules: $RB_\infty[1]$ is the right shift over dark or light, and $RB_\infty[2]$ is the identity rule over red or blue. Therefore, under $RB_\infty$, any cell that starts out red or blue remains that color; and, under almost all initial conditions, any cell's neighborhood of radius $r$ contains, after some finite number of generations, any dark-light pattern of length $2r + 1$.

The rules in the $RB$ sequence are formed as follows. Rule $RB_n$ is a rule of radius $n$ that behaves like the limit rule except when a cell's neighborhood of radius $n$ is all light. At that time, the cell, if it is red, changes to blue. Now,

$$\lim_{n \to \infty} d(RB_n, RB_\infty) = \lim_{n \to \infty} \frac{2^{2n+1}}{4^{2n+1}} = 0, \tag{32}$$

so the sequence converges to the limit rule. However, for almost all initial conditions, each cell on a doubly infinite lattice will eventually become blue. Thus, the behavior of each rule in the sequence differs significantly from the limit rule.

This example of discontinuity differs from the first in two respects: anomalous table entries are not used significantly often; and the limit language of each rule in the sequence ( ((dark, blue) + (light, blue))* ) is invariant under the limit rule. It can be shown that if a sequence of rules meets the first of these conditions, it must meet the second.

To establish this, some definitions must be made. Let the *transient language* of a rule $T(R)$ be those members of $\Sigma^*$ that must eventually disappear, under repeated applications of $R$, from any finite portion of a doubly infinite lattice, under almost all initial conditions. Let the *nontransient language* of a rule $N(R) = \Sigma^* - T(R)$. Note that $N(R)$ is not the same as the limit language of $R$. For example, consider rule 128, described above. The string 111 is part of the limit language of this rule, since it has a preimage arbitrarily many generations back. However, unless the initial conditions are all ones, 111 will eventually disappear from any finite portion of the lattice.

**Theorem 10.** *Let $R_1$ and $R_2$ be cellular automaton rules over the same alphabet $\Sigma^*$. Let $R_1'$ and $R_2'$ be equivalent to $R_1$ and $R_2$, but having equal radii. Suppose that, under almost all initial conditions, on any finite area of the doubly infinite lattice, $R_2'$ uses table entries different from $R_1'$ only a finite number of times. Then $N(R_2)$ is invariant under $R_1$.*

**Proof.** Since, on a doubly infinite lattice, rules operate the same as their equivalents, it is enough to consider rules of equal radii.

Suppose $\exists$ $w$ not in $N(R_2)$ such that $w = R_1(x)$, $x$ in $N(R_2)$. Under almost all initial conditions, $R_2$ will continue to produce $x$ indefinitely on at least some finite sections of the lattice. Since table entries differing from $R_1$ are only used transiently, eventually, on each finite section of the lattice producing $x$ indefinitely, $w$ will also be produced indefinitely. This contradicts the assumption that $w$ is not in $N(R_2)$.

On the other hand, suppose $\exists$ $x$ in $N(R_2)$ that does not have a preimage under $R_1$. Eventually, in each finite section of the lattice, only table entries the same as $R_1$ will be used. Therefore $x$ cannot be produced indefinitely in any finite section of the lattice, contradicting the assumption that $x$ is in $N(R_2)$. ∎

**Corollary.** *Let a sequence of cellular automaton rules $R_1, \ldots, R_n, \ldots$ converge to limit rule $R_0$. Suppose that, under almost all initial conditions and for all $k$, $R_k$ uses table entries differing from $R_0$ only transiently (as described in the above theorem). Then, for all $k$, $N(R_k)$ is invariant under $R_0$.*

## 5. Conclusion

This paper has presented algorithms for determining whether or not a regular language is invariant under the application of a cellular automaton rule. A metric has been devised for the set of all cellular automaton rules over a given alphabet; and the topological properties of various classes of rules, under this metric, have been investigated. Lastly, an investigation has begun into discontinuities of behavior in cellular automaton rule space. From numerical experiments it appears that there are at least two types of such discontinuities.

## Appendix A: Rule 31 random number generator

```
static unsigned int x=1;
unsigned int random() {
  static unsigned int first=1;
  static float f;
  static int i;
  if (first) {/* transient */
     first = 0;
     for(i=0;i<100;i++)
       x = (x | ((x<<1)|(x>>31))) ^ ((x>>1)|(x<<31));
```

```
  }
  x = (x | ((x<<1)|(x>>31))) ^ ((x>>1)|(x<<31));
  return(x);
}
```


## Acknowledgments

I am grateful to Norman Packard and David Muller for extensive suggestions and discussion. I would also like to thank Larry Dornhoff, Chris Hartman, and Jay Obermark for helping me with the computer experiments.

This paper is dedicated to the memory of D. B. Levine and S. B. C. Levine.


## References


[1] J. E. Hanson and J. P. Crutchfield, "The Attractor-Basin Portrait of a Cellular Automaton," Santa Fe Institute preprint 91-02-012 (1991).

[2] G. A. Hedlund, "Endomorphisms and Automorphisms of the Shift Dynamical System," *Mathematical Systems Theory*, **3** (1969) 320.

[3] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Formal Languages and Computation* (Reading, MA, Addison-Wesley, 1979).

[4] J. Milnor, "Notes on Surjective Cellular Automaton-Maps," Institute for Advanced Study notes (June 1984).

[5] S. Wolfram, "Computation Theory of Cellular Automata," *Communications in Mathematical Physics*, **96** (1984) 15–57.

[6] S. Wolfram, "Statistical Mechanics of Cellular Automata," *Reviews of Modern Physics*, **55** (1983) 601–644.