*Complex Systems* 4 (1990) 119–129

# Recursive Cellular Automata Invariant Sets

**Lyman P. Hurd**
*Lab for Plasma Research, University of Maryland,*
*College Park, MD 20742, USA*

**Abstract.** Subshifts of bi-infinite words give rise to languages in a natural way. Three invariant subshifts associated with cellular automata dynamics are considered: the periodic set, non-wandering set, and limit set. Examples have been given (see [8,12]) of cellular automata whose invariant sets give rise to nonrecursive languages. In this paper, examples are given of cellular automata whose invariant sets correspond to regular, context-free, and context-sensitive languages. It is proved that a cellular automaton has a subshift of finite type limit set only if its images stabilize. As a corollary, only mixing subshifts of finite type can occur as cellular automaton limit sets.

## 1. Introduction

Subshifts of the full shift on $k$ symbols give rise to languages in a natural way [5]. Of particular interest are those subshifts that arise in the dynamics of shift-endomorphisms, cellular automata. In [8], a series of examples were given of cellular automata for which the limit set (intersection of forward images) gave rise to nonregular languages. In particular an example was given for which the limit language was provably nonregular, and one for which it was provably not context-free. This paper strengthens these results by giving a complete description of cellular automaton limit sets giving rise to context-free and context-sensitive languages, and extends these results to the periodic and non-wandering sets.

It was also shown in [8] that the general question of whether a given string appeared in the limit language of a cellular automaton was undecidable. These results were strengthened in [10,12] to give an explicit construction of a cellular automaton whose limit set gave rise to a language that was not recursively enumerable. An independent proof was given by Culik, Pachl, and Yu (see [2]). For a general survey of these approaches to the analysis of cellular automaton behavior, see [4].

Section 2 reviews the definitions of cellular automata as well as defining the periodic set, non-wandering set, and limit set. Definitions from language theory can be found in [7]. Section 3 gives a series of examples of rules whose invariant sets have increasing complexity. Section 4 reviews the results of [12] without proofs. And Section 5 discusses the significance of these results.

## 2. Definitions

### 2.1 Cellular Automata

The full shift on $k$ symbols, $S^{\mathbf{Z}}$, is the set of functions from the integers to a set of $k$ elements, $S$. Equivalently elements of $S^{\mathbf{Z}}$ are doubly infinite words in the symbols from $S$ with a fixed base symbol. This space is topologized by the metric:

$$d(x,y) = \sum_{i=-\infty}^{\infty} \delta(x_i, y_i)\, 2^{-|i|}$$

where $\delta(a,b) = 0$ if $a = b$ and 1 otherwise.

The full shift is acted upon by a homeomorphism $\sigma$ (the left shift map) defined by $\sigma(x)_i = x_{i+1}$.

**Definition 2.1.** *A subshift is a closed, shift-invariant subset of $S^{\mathbf{Z}}$.*

The natural maps to consider on these spaces, are continuous maps that commute with the shift.

**Definition 2.2.** *A cellular automaton is a continuous function $F : S^{\mathbf{Z}} \to S^{\mathbf{Z}}$ that commutes with $\sigma$.*

Cellular automata are specified by giving a *local* function

$$f : S^{2r+1} \to S$$

and defining a *global* function $F : S^{\mathbf{Z}} \to S^{\mathbf{Z}}$ by:

$$F(x)_i = f(x_{i-r}, \ldots, x_{i+r})$$

The integer $r > 0$ is called the *radius* of the cellular automaton rule. The Curtis–Hedlund–Lyndon theorem states that all cellular automata arise in this fashion (see [6]).

Given a finite string in the symbols from $S$, and a starting position, $n$, one can construct the *cylinder set* $\mathrm{Cyl}(s)$ defined by:

**Definition 2.3.** $\mathrm{Cyl}(s_1 \ldots s_n) = \{c \in S^{\mathbf{Z}} \mid c_i = s_i$ *for* $-[(n-1)/2] \leq i \leq [n/2]\}$ *where* $[x]$ *denotes the greatest integer in* $x$.

The cylinder sets are open and closed, and form a basis for the topology of $S^{\mathbf{Z}}$. The language associated with a subshift is the set of finite blocks occurring in configurations from $K$. Languages arising from subshifts have been studied in their own right (see [1,14,15,16]. A complete characterization of languages arising from subshifts is given in [4].

**Definition 2.4.** $\mathcal{L}(K) = \{s \in S^* \mid \mathrm{Cyl}(s) \cap K \neq \emptyset\}$ *where* $S^*$ *is the set of all finite strings in the symbols from* $S$.

Since $K$ is shift-invariant, $\mathcal{L}(K)$ is the set of all finite blocks occurring in configurations from $K$. The usefulness of this definition arises from the fact that the set of finite strings completely determines the subshift.

**Lemma 2.1.** *If* $K_1, K_2 \subseteq S^{\mathbf{Z}}$ *are subshifts,* $\mathcal{L}(K_1) = \mathcal{L}(K_2)$ *implies* $K_1 = K_2$.

**Proof:** This lemma was stated in [18] and proved in [8]. ∎

## 2.2 Subshifts of Finite Type and Sofic Systems

An important class of subshifts are described by giving a finite list of blocks that *may not* appear as subblocks in a configuration. Such a subshift is said to be of *finite type*.

Formally, if $w_1, \ldots, w_n \in S^*$ is a finite list of words, the *subshift of finite type*, they generate is the largest subshift $K$ with the property that $w_i \notin \mathcal{L}(K)$ for $1 \leq i \leq n$. In general, a subshift of finite type may arise from different lists of excluded blocks.

Every subshift of finite type satisfies the property that $\mathcal{L}(K)$ is a regular language (see [18,19,3,10]), however not every subshift that generates a regular language is of this type. A subshift that is the image under a cellular automaton map of a subshift of finite type is called a *sofic system* (see [18]). The finite time image of a cellular automaton is a sofic system that satisfies an additional condition known as mixing.

A subshift is *mixing* if every block that occurs once in a configuration occurs twice in some other configuration. In other words, if $x \in \mathcal{L}(K)$ then $xyx \in \mathcal{L}(K)$ where $y$ is an arbitrary block (possibly empty). If $K$ is mixing, so is $F(K)$ where $F$ is a cellular automaton.

It can be shown that a subshift $K$ is sofic if and only if $\mathcal{L}(K)$ is a regular language (see [10]).

## 2.3 Invariant Subshifts

In order to be of interest dynamically, it is essential that the various generalizations of sofic systems actually appear in the dynamics of shift-invariant maps (cellular automata). This paper deals with three subshifts that arise from the dynamics of a cellular automaton rule $F$, and that are studied in topological dynamics.

### 2.3.1 The Limit Set

Cellular automata are not generally invertible. Most rules form a decreasing sequence of images

$$S^{\mathbf{Z}} \supseteq F(S^{\mathbf{Z}}) \supseteq \ldots$$

The *limit set*, $\Lambda(F)$, of a cellular automaton is the intersection of all forward images.

$$\Lambda(F) = \bigcap_{i=0}^{\infty} F^i(S^{\mathbf{Z}})$$

The limit set is the set of all $c \in S^{\mathbf{Z}}$ that have an infinite sequence of pre-images. Often the easiest way to prove that a configuration is in $\Lambda(F)$ is to construct such a sequence.

### 2.3.2 The Periodic Set

A configuration is periodic if $F^p(x) = x$ for some $p > 0$. The set of periodic points of a cellular automata is not generally closed (periodic points for the left shift map are dense in the whole space). Therefore, the *periodic set*, $\Pi(F)$, of a cellular automaton is defined to be the closure of the set of $c \in S^{\mathbf{Z}}$ such that $F^p(c) = c$ for some $p > 0$. Note that taking the closure does not introduce any new finite blocks.

### 2.3.3 The Non-Wandering Set

A weaker condition than periodicity is the condition that a point be non-wandering (see [17]). A point $x$ is *wandering* if there is a neighborhood of $x$ which all orbits leave and never return (in particular $x$ cannot be periodic). The non-wandering set is the complement of the wandering points.

**Definition 2.5.** *If $F : X \to X$ is a continuous map of a compact space, the non-wandering set of $F$, denoted $\Omega(F)$, consists of all points $x \in X$ such that for every neighborhood $U$ of $x$, there exists an integer $n \geq 1$ such that $F^{-n}(U) \cap U \neq \emptyset$.*

It follows from the definition that $\Omega(F)$ is a subshift and $\Pi(F) \subseteq \Omega(F) \subseteq \Lambda(F)$. In practice these inequalities can be strict (see [11]).

The following technical lemma is helpful in actually calculating the set of blocks arising in non-wandering configurations. Taking $U$ in the above definition to be a cylinder set containing $x$ gives the following characterization.

**Lemma 2.2.** *A block $w$ occurs as a subblock of a configuration in the non-wandering set of a cellular automaton $F$ if there exist configurations $c, c'$ such that $F^n(c) = c'$ for some $n > 0$ and $w$ occurs as a subblock of both $c$ and $c'$ in the same position.*

## 3. Recursive Invariant Sets

### 3.1 Subshift of Finite Type Invariant Sets

The simplest form of subshift we have considered is that of a subshift of finite type. Having $\Lambda(F)$ be a subshift of finite type poses severe restrictions on the behavior of a rule. In fact, it implies that there exists a time step $N$ for which the images stabilize ($F^n(S^{\mathbf{Z}}) = F^N(S^{\mathbf{Z}})$) for all $n \geq N$ with equality holding on the set level, not necessarily pointwise.

The proof of this fact hinges on the following easy topological lemma:

**Lemma 3.1.** *For any cellular automaton rule $F$, if $s \in \mathcal{L}(F^i(S^{\mathbf{Z}}))$ for all $i \geq 0$, then $s \in \mathcal{L}(\Lambda(F))$.*

**Proof:** Observe that

$$\mathrm{Cyl}(s) \cap \Lambda(F) = \bigcap_{i=0}^{\infty} (\mathrm{Cyl}(s) \cap F^i(S^{\mathbf{Z}}))$$

The right-hand side is a decreasing intersection of nonempty compact sets, and therefore is nonempty. ∎

**Theorem 3.2.** *If $\Lambda(F)$ is a subshift of finite type, then there exists $n \geq 0$ such that $F^n(S^{\mathbf{Z}}) = \Lambda(F)$.*

**Proof:** A subshift of finite type is specified by a finite list of excluded blocks. By lemma 3.1, each of these blocks must have been excluded by some finite time. Let $n$ be the maximum of these exclusion times. ∎

This theorem gives one of the few criteria we have to state that a subshift cannot occur as the limit set of a cellular automaton. Since the property of being mixing is preserved by finite iteration of a cellular automaton map, it follows immediately that:

**Corollary 3.3.** *If $\Lambda(F)$ is a subshift of finite type, it is mixing.*

An example of this behavior is found in [19]. The rule $F_1$, (elementary rule 236 in Wolfram's classification scheme) is defined on the set $\{0,1\}^{\mathbf{Z}}$ and is generated by the local function $f(1,0,1) = 1$ and otherwise $f(x,y,z) = y$.

The image $F_1(S^{\mathbf{Z}})$ consists of those configurations not containing the substring 101. This is also the set of points fixed by $F_1$. It is immediate that $F_1(S^{\mathbf{Z}}) = \Lambda(F_1) = \Omega(F_1) = \Pi(F_1)$. In this case, $\Lambda(F_1)$ is a subshift of finite type (the fixed points of a cellular automaton always form a subshift of finite type) given by excluding the block 101.

If the limit set is a subshift of finite type, it must be equal to one of the map's finite time images. However, it is not known whether stabilizing at a finite time guarantees that the limit set will be a subshift of finite type.

### 3.2 Regular (Sofic) Invariant Sets

Rule $F_2$ is Wolfram's Elementary Rule 128, which is given by the local function $f(1,1,1) = 1$ and $f(x,y,z) = 0$ otherwise. At each time step $t$, the language $\mathcal{L}(F_2^t(S^{\mathbf{Z}}))$ is the subshift of finite type all configurations not containing strings of the form $10^i1$ for $i \leq 2t$. In particular $F_2^m(S^{\mathbf{Z}}) \neq F_2^n(S^{\mathbf{Z}})$ for $m \neq n$. The limit set consists of all strings that do not contain any string of the form $10^i1$ for any $i > 0$. Therefore $\mathcal{L}(\Lambda(F)) = \{0^*1^*0^*\}$.

This set has an infinite number of distinct excluded blocks, and is a sofic system but *not* a subshift of finite type. Furthermore it is not mixing (the block 010 can occur once but not twice).

The periodic set, $\Pi(F)$ for this rule consists of the two fixed points, $1^*$ and $0^*$. Intuitively $1^*$ is a repelling fixed point and $0^*$ an attracting one.

In the case of rule $F_2$ we observe that neither of the strings, 01 and 10 meet the criterion of lemma 2.2. This implies that the only configurations that can be in the non-wandering set are the constant ones. In this case

$$\Omega(F_2) = \Pi(F_2) = \{0^*, 1^*\}$$

### 3.3 Context-Free Invariant Sets

To construct a rule whose invariant sets correspond to a context-free language, one simulates left and right-moving particles interacting with walls in a quiescent background. By allowing the particles to bounce off the walls only in symmetric pairs, one can set up resonances such as those found in the language $a^nba^n$ that is a classical example of a nonregular context-free language (see [7]).

The original idea behind this rule was suggested to me by Danny Hillis. It is similar to an example in [8]. $F_3$ has $S = \{O, W, r, R, l, L\}$ and a local rule depending on sites up to two sites away (i.e., $r = 2$). This rule has five particles that interact in a quiescent background. Walls, $W$, stay stationary for all time; right-moving gliders $r, R$, and left-moving gliders move at unit speed (one site per iteration) in their respective directions. Colliding gliders, or gliders colliding with a wall annihilate. However, if a glider of type $R$ hits a wall from the left and a glider of type $L$ hits a wall on the right *simultaneously*, they bounce to become gliders of type $l$ and $r$ respectively.

The rule table for rule $F_3$ is illustrated in table 1, and the evolution from a sample initial state is given in figure 1.

**Theorem 3.4.** *$\Lambda(F_3) = K_1 \cup K_2$ where*

$$\begin{aligned} K_1 &= \{(r+R+O)^*(W+O)^*(l+L+O)^*\} \\ K_2 &= \{(r+R+O)^* lO^iWO^ir(l+L+O)^*\} \end{aligned}$$

**Proof:** The proof to this theorem and theorem 3.7 consist of straightforward enumerations of cases. Details of calculations of this sort can be found in [10]. ∎

**Corollary 3.5.** *$\mathcal{L}(\Lambda(F_3))$ is a context-free language.*

**Proof:** Since $\mathcal{L}(K_3)$ is a regular language and $\mathcal{L}(K_2)$ is a context-free language, $\mathcal{L}(\Lambda(F_3)) = \mathcal{L}(K_2) \cup \mathcal{L}(K_3)$ is context-free.

To show that the limit language is not regular uses the fact that the intersection of two regular languages is regular. Consider the intersection of $L$ with the language $R_1$ generated by the regular expression $lO^*WO^*r$. The intersection is precisely $\{lO^iWO^ir\}$, which by the pumping lemma for regular languages (see [7]), is a nonregular context-free language. ∎

**Theorem 3.6.** $\Pi(F_3) = \Omega(F_3) = \Lambda(F_3)$.

**Proof:** Since $\Pi(F) \subseteq \Omega(F) \subseteq \Lambda(F)$, it suffices to show that periodic points are dense in $K_1$ and $K_2$. Equivalently, we need to show that every string in $\mathcal{L}(\Lambda(F))$ occurs in some periodic configuration.

Every string in $\mathcal{L}(K_1)$ occurs in a periodic configuration of the form: $c_r^* c_w c_l^*$ where $c_r \in \{(r+R+O)^*\}$, $c_l \in \{(l+L+O)^*\}$, and $c_w \in \{(W+O)^*\}$ are fixed strings.

Every string in $\mathcal{L}(K_2)$ occurs in a periodic point of the form:

$$(c_r O^{2i+1} R)^* c_r lO^iWO^irc_l(LO^{2i+1}c_l)^*$$

∎

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| . | . | W | . | . | $\mapsto$ | W | |
| . | . | R | W | L | $\mapsto$ | l | |
| R | W | L | . | . | $\mapsto$ | r | |
| . | r | x | y | . | $\mapsto$ | r | (x $\neq$ W x,y $\neq$ l,L |
| . | R | x | y | . | $\mapsto$ | R | (x $\neq$ W x,y $\neq$ l,L |
| . | x | y | l | . | $\mapsto$ | l | (y $\neq$ W x,y $\neq$ r,R |
| . | x | y | L | . | $\mapsto$ | L | (y $\neq$ W x,y $\neq$ r,R |
| otherwise | | | | | $\mapsto$ | O | |

(. = any symbol)
highest applicable rule takes precedence

Table 1: Rule $F_3$.

```
rl   WR W r l  r LWRl       R      W      L      Wl  Wr W    lr  R  L  rl
       W RW         W          R     W     L       W   W rW   l  r  RL
       W  W         W           R    W    L        W   W  W  l    r
       W  W         W            R   W   L         W   W  W l      r
       W  W         W             R  W  L          W   W  Wl        r
       W  W         W              R W L           W   W  W          r
       W  W         W               RWL            W   W  W           r
       W  W         W               lWr            W   W  W            r
       W  W         W              l W r           W   W  W             r
       W  W         W             l  W  r          W   W  W              r
```

Figure 1: Evolution of rule $F_3$ from a sample initial condition.

### 3.4 Context-Sensitive Invariant Sets

The simplest example of a language that is context-sensitive but not context-free is $a^n b^n c^n$. In this case we allow correlations among four values by allowing particles to go at two speeds. Once again particles incident on a wall bounce only under special conditions.

The rule $F_4$ has $S = \{O, W, r, R, l, L\}$ as above, but its local rule depends on sites up to four away. Again $W$ represents a wall. The two varieties of glider represent different speeds. Gliders $R$ and $L$ travel at two sites per iteration; $r$ and $l$ travel at unit speed in their respective directions. Gliders cannot pass through one another; if a glider collides with another glider or a wall it disappears.

There is one exception to this rule. If a wall is hit by two slow gliders on the left and on the right in the form $rrWll$, they yield $LlWrR$. This interaction allows correlation among four spaces that gives rise to context-sensitive but not context-free limiting behavior.

The rule table for $F_4$ is given in table 2, and its evolution from a sample initial state is illustrated in figure 2.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| . | . | . | . | W | . | . | . | . | $\mapsto$ | W | |
| r | r | W | l | l | . | . | . | . | $\mapsto$ | R | |
| . | r | r | W | l | l | . | . | . | $\mapsto$ | r | |
| . | . | . | . | r | r | W | l | l | $\mapsto$ | L | |
| . | . | . | r | r | W | l | l | . | $\mapsto$ | l | |
| . | . | a | r | b | c | d | . | . | $\mapsto$ | r | (a$\neq$R b$\neq$W,l,L c$\neq$l,L d$\neq$L) |
| . | . | R | a | b | c | d | . | . | $\mapsto$ | R | (a$\neq$r,W,l,L b$\neq$W,l,L c$\neq$l,L d$\neq$L) |
| . | . | a | b | c | l | d | . | . | $\mapsto$ | l | (d$\neq$L c$\neq$W,r,R b$\neq$r,R a$\neq$R) |
| . | . | a | b | c | d | L | . | . | $\mapsto$ | L | (d$\neq$l,W,r,R c$\neq$W,r,R b$\neq$r,R a$\neq$R) |
| otherwise | | | | | | | | | $\mapsto$ | O | |

(. = any symbol)

highest applicable rule takes precedence

Table 2: Rule $F_4$.

```
R   r   L      l  W        rr    W    ll       r     Rl        L
  R  rL      l    W         rr   W   ll         r            L
            l     W          rr  W  ll           r         L
           l      W           rr W ll             r      L
          l       W            rrWll               r   L
         l        W            LlWrR                rL
        l         W          L l W r R
       l          W        L  l  W  r  R
      l           W      L   l   W   r   R
     l            W    L    l    W    r    R
```

Figure 2: Evolution of rule $F_4$ from a sample initial condition.

**Theorem 3.7.** $\Lambda(F_4) = S_1 \cup S_2$ *where*

$$\begin{aligned} S_1 &= \{(R+O)^*(r+O)^*(W+O)^*(l+O)^*(L+O)^*\} \\ S_2 &= \{(R+O)^*(r+O)^*LO^ilO^iWO^irO^iR(l+O)^*(L+O)^*\} \end{aligned}$$

**Corollary 3.8.** *$\mathcal{L}(\Lambda(F_4))$ is a context-sensitive language.*

**Proof:** Since $\mathcal{L}(S_1)$ is a regular language and $\mathcal{L}(S_2)$ is a context-sensitive language, $\mathcal{L}(\Lambda(F_4))$ is context-sensitive. The fact that it is not context-free follows ¿from the fact that the intersection of $\mathcal{L}(\Lambda(F_4))$ with the regular language $\{LO^*lO^*WO^*rO^*R\}$ is $\{LO^ilO^iWO^irO^iR\}$, a language that violates the pumping lemma for context-free languages.

No block of the form $rO^*R$ or $LO^*l$ can occur in the limit set. From this it follows that no periodic point can have a block of the form $RO^*r$ or $lO^*L$. ∎

**Theorem 3.9.** $\Pi(F_4) = \Omega(F_4) = T_1 \cup T_2$ *where*

$$\begin{aligned} T_1 &= \{((r+O)^* + (R+O)^*)(W+O)^*((l+O)^* + (L+O)^*)\} \\ T_2 &= \{(r+O)^*LO^ilO^iWO^irO^iR(l+O)^*\} \end{aligned}$$

## 4. Non-R.E. Invariant Sets

These results appear elsewhere (see [12]) and are sketched here for the sake of completeness. First note that there is an upper bound on the complexity of a cellular automaton limit language given by a semi-procedure that halts only if a string is not in this set.

**Theorem 4.1.** *For every cellular automata $F : S^{\mathbf{Z}} \to S^{\mathbf{Z}}$, the complement of the limit language, $S^* - \mathcal{L}(\Lambda(F))$, is recursively enumerable.*

Given this constraint, however, there exist arbitrarily complicated cellular automaton languages. More exactly, given any language whose complement is recursively enumerable, one can construct a cellular automaton whose limit language yields the chosen language after intersection with a regular language, and $\varepsilon$-limited homomorphism.

**Theorem 4.2.** *If $L \subseteq A^*$ is a language whose complement is recursively enumerable, there exists a cellular automaton $F : S^{\mathbf{Z}} \to S^{\mathbf{Z}}$, a regular language $R \subseteq S^*$, and a homomorphism $\phi : S^* \to A^*$ such that:*

$$\phi(\mathcal{L}(\Lambda(F)) \cap R) = L$$

The procedure of intersection with a regular language, and homomorphisms both preserve the class of recursively enumerable languages. The proof of this theorem, coupled with the existence of languages that are not recursively enumerable although their complements are, yield an example of a cellular automaton whose limit language fails to be recursively enumerable.

**Corollary 4.3.** *There exists a cellular automaton $F_u : S^{\mathbf{Z}} \to S^{\mathbf{Z}}$ such that $\mathcal{L}(\Lambda(F_u)) \subseteq S^*$ is not a recursively enumerable language.*

Temporally periodic orbits of a given period have a particularly simple description. Using this procedure for each time step in turn gives rise to a procedure for determining if a given block occurs in a temporally periodic configuration.

**Theorem 4.4.** *If $F$ is any cellular automaton rule, $\mathcal{L}(\Pi(F))$ is recursively enumerable.*

Again, given this constraint, the behavior exhibited by the periodic set can be quite complex. Basically a cellular automaton is set up to simulate a Turing machine that upon halting starts the computation all over again.

**Theorem 4.5.** *If $L \subseteq A^*$ is any recursively enumerable language, there exists a cellular automaton rule $F : S^{\mathbf{Z}} \to S^{\mathbf{Z}}$, a regular language $R \subseteq S^*$, and a homomorphism $\phi : S^* \to A^*$ such that:*

$$\phi(\mathcal{L}(\Pi(F)) \cap R) = L$$

## 5. Discussion

The finite time image of a cellular automaton map is always described by a regular language. Often, however, one is interested in invariant sets whose definition involves longer time scales. In particular the three shifts considered here, the periodic set, non-wandering set, and limit set all require knowledge of histories for all time. In these cases, although the dynamics of a cellular automaton always have a finite description, their limiting behavior may be undecidable.

The correspondence between languages and subshifts described by lemma 2.1 links the dynamics of continuous maps on a particular compact metric space, with formal language theory. The series of examples above serve to show that the dynamical systems can exhibit much of the same range of complexity found in language theory.

The examples here have all been by nature synthetic. The inverse problem would be taking a cellular automaton and determining the complexity of its invariant sets. Although some progress has been made (see [9]), many of Wolfram's elementary rules have yet to have their invariant sets described. It is unknown whether one of these rules has a nonregular limit language. It is a corollary of recent work of Kari [13] that determining the language complexity of the limit set for an arbitrary cellular automaton is undecidable.

## Acknowledgments

I am grateful for discussions with Karel Culik, Mike Hurley, Jarkko Kari, John Milnor, and Sheng Yu. This work was supported by DARPA under the ACMP program.

## References

[1] L. Boasson and M. Nivat, "Adherences of languages", *Journal of Computer System Sciences*, **20** (1980) 285–309.

[2] K. Culik II, J. Pachl, and S. Yu, "On the limit sets of cellular automata," *SIAM J. Comput.*, **18:4** (1989) 831–842.

[3] K. Culik II and S. Yu, "Cellular automata, $\omega\omega$-regular sets, and sofic systems," *Discrete Applied Mathematics*, to appear.

[4] K. Culik II, L.P. Hurd, and S. Yu, "Computation theoretic aspects of cellular automata" *Physica D*, submitted.

[5] K. Culik II, L.P. Hurd, and S. Yu, "Formal language aspects of global cellular automaton behavior," *Physica D*, submitted.

[6] G.A. Hedlund, "Transformations commuting with the shift," in *Topological Dynamics*, J. Auslander and W.G. Gottschalk, eds. (Benjamin, 1968) 259–289; and "Endomorphisms and automorphisms of the shift dynamical system," *Math. Syst. Theor.*, **3** (1969) 320–375.

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

[8] L.P. Hurd, "Formal language characterizations of cellular automaton limit sets," *Complex Systems*, **1** (1987) 69–80.

[9] L.P. Hurd, "Regular language complexities," a table in *Theory and Applications of Cellular Automata* (World Scientific, 1986) 523–526.

[10] L. P. Hurd, "The application of formal language theory to the dynamical behavior of cellular automata," doctoral thesis, Department of Mathematics, Princeton University, June 1988.

[11] L.P. Hurd, "The non-wandering set of a CA map," *Complex Systems*, **2** (1988) 549–554.

[12] L.P. Hurd, "Nonrecursive cellular automata invariant sets," *Complex Systems*, to appear.

[13] J. Kari, "Rice's theorem for limit sets of cellular automata", in preparation.

[14] M. Nivat, "Sur les ensembles de mots infinis engendrés par une grammaire algébrique," *RAIRO Informatique Théorique*, **12** (1978) 259–278.

[15] M. Nivat and D. Perrin, "Ensembles reconnaissables des mots biinfinis," *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing* (1982) 47–59.

[16] A. Restivo, "Finitely Generated Sofic Systems," Research Report, Dipartimento di Matematica e Applicazioni, Universita di Palermo (1986).

[17] P. Walters, *An Introduction to Ergodic Theory* (Springer-Verlag, 1982).

[18] B. Weiss, "Subshifts of finite type and sofic systems," *Monatshefte fur Mathematik*, **77** (1973) 462–474.

[19] S. Wolfram, "Computation theory of cellular automata," *Commun. Math. Phys.*, **96** (1984) 15–57.