Languages not recognizable in real time by one-dimensional cellular automata

Katsuhiko Nakamura

School of Science and Engineering, Tokyo Denki University, Hatoyama-machi, Saitama-ken 350-0394, Japan

Received 22 December 2004; received in revised form 10 June 2007
Available online 16 June 2007

Abstract

This paper discusses real-time language recognition by one-dimensional cellular automata (CA), focusing on limitations of the parallel recognition power. We investigate language recognition of strings containing binary representations $B(|w|)$ of their own lengths. It is shown that:

1. The language $L_X = \{w \in \{0, 1\}^+: w$ contains the binary number(s) $B(|w|)\}$ is recognizable by CA in linear time, but is not recognizable in real time; and
2. The class of languages that are recognizable by CA in real time is not closed under concatenation and not closed under reversal.

These results are solutions to the problems posed by Smith III in [A.R. Smith III, Real-time language recognition by one-dimensional cellular automata, J. Comput. System Sci. 6 (1972) 233–253].

© 2007 Elsevier Inc. All rights reserved.

Keywords: CA; OCA; Parallel language recognition; Real-time recognition; Closure under concatenation

1. Introduction

Language recognition by bounded cellular automata (CA) was first investigated by Kasami and Fujii [7] and Cole [2] in the late 1960s. In studies of language recognition by CA, parallel recognition by deterministic CA, including real-time and linear-time recognition, has been the most important subject. Smith III [10] first discussed real-time language recognition by one-dimensional CA in 1972. In 1980, Dyer [3] first discussed real-time language recognition power of one-way CA (OCA). Real-time language recognition is especially important, since it is likely that languages are recognized in real time in the human mind. Although parallel language recognition by CA has been investigated for more than three decades, many important problems still remain unsolved.

This paper discusses parallel recognition by one-dimensional deterministic bounded CA, and gives some solutions to the problems on limitations of parallel recognition power.
1.1. CA, OCA and their languages

A cellular automaton (CA) is a system \( S = (K, \#, f, A) \), where:

- \( K \) is a finite set of states;
- \( \# \in K \) is the boundary state;
- \( f \), a transition function, is a mapping from \( K \times K \times K \) into \( K - \{\#\} \); and
- \( A \subseteq K \) is a set of accepting states.

A one-way cellular automaton (OCA) is a CA \( S = (K, \#, f, A) \) such that there exists a function \( f' : K \times K \rightarrow K \) with \( f(x, y, z) = f'(x, y) \) for all \( x, y, z \in K \).

A configuration is a string \( \#u\# \) of states with \( u \in (K - \{\#\})^+ \). The transition function \( f \) is extended to configurations by

\[
f(\#a_1a_2\ldots a_n\#) = \#b_1b_2\ldots b_n\#
\]

where \( b_i = f(a_{i-1}, a_i, a_{i+1}) \) for all \( i \in \{1, 2, 3, \ldots, n\} \), and \( a_0 = a_{n+1} = \# \). The function \( f^n \) of configurations is recursively defined by \( f^0(c) = c \) and \( f^n(c) = f(f^{n-1}(c)) \), for all configurations \( c \) and \( n \geq 0 \).

A CA \( S = (K, \#, f, A) \) recognizes a language \( L \subseteq \Sigma^+ \) in a time \( t \) (by the rightmost cell), if and only if \( \Sigma \subseteq K \) and for all \( w \in \Sigma^+ \), \( |w| \geq 2 \) (Fig. 1),

\[
w \in L \iff f^t(\#w\#) \in \{(\#)K^+A\}.
\]

The CA \( S \) recognizes \( L \) in real time, if and only if \( t = |w| - 1 \). The CA \( S \) recognizes \( L \) in linear time, if and only if there is a constant \( c \geq 1 \) with \( t = c|w| \).

The computation speed can be increased as shown by the speed-up lemma [11]. By a \( k \) times speed-up, the time \( c|w| = cn \) of linear-time recognition can be reduced to almost real time, \( n(1 + c/k) \) for any integer \( k \geq 2 \). The real time recognition is still important, as Smith [10] stated that “the cost of a speed-up by a factor of \( k \), in terms of the size of the state set, increases exponentially with \( k \).”

For real-time recognition, we can write \( f^t(a_1a_2\ldots a_n\#) = b_{i+1}\ldots b_n\# \) for all \( i \in \{1, 2, \ldots, n - 1\} \). For real-time recognition by an OCA, we also write \( f^t(a_1a_2\ldots a_n) = b_{i+1}\ldots b_n \).

The language recognized by the leftmost cell, called the left CAL or left OCAL, is defined similarly, and is equivalent to the reversal of the language recognized by the rightmost cell. Since this paper mainly deals with acceptance by the rightmost cell, we use the simplified terms CAL and OCAL to refer the “right CAL” and “right OCAL.” Note that in some literatures [5,6] left CAL is the standard and simply called CAL.

1.2. Previous results

The results of previous researches closely related to this paper are summarized as follows, where \( \mathcal{C}(L) \) denotes the class of languages \( L \).

1. \( \mathcal{C}(\text{CAL}) \) is equal to \( \mathcal{C}(\text{deterministic context sensitive language}) \) [7,10]. \( \mathcal{C}(\text{OCAL}) \) includes \( \mathcal{C}(\text{context free language}) \) [7].

2. \( \mathcal{C}(\text{real-time OCAL}) \), as well as \( \mathcal{C}(\text{real-time CAL}) \), includes non-context-free languages such as \( \{0^n1^n2^n : n \geq 1\} \) [3,10]. \( \mathcal{C}(\text{real-time CAL}) \) includes complex languages such as \( \{1^n : n \text{ is a prime number}\} \) [4,10].

3. Both \( \mathcal{C}(\text{real-time OCAL}) \) and \( \mathcal{C}(\text{real-time CAL}) \) are closed under set operations [10].
(4) $\mathcal{C}$ (real-time OCAL) is closed under reversal, hence $\mathcal{C}$ (real-time OCAL) = $\mathcal{C}$ (real-time left OCAL) [1]. $\mathcal{C}$ (linear-time CAL) is also closed under reversal [10].
(5) $\mathcal{C}$ (real-time CAL) = $\mathcal{C}$ (linear-time left OCAL) and $\mathcal{C}$ (real-time left CAL) = $\mathcal{C}$ (linear-time OCAL) [1,12].

Problems concerning limitations of the real-time language recognition power of CA and OCA are especially important. In 1984, Choffrut and Culik II [1] proved that $\mathcal{C}$ (real-time OCAL) is a proper subclass of $\mathcal{C}$ (real-time CAL) by showing that the language $\{1^{2n}: n \geq 0\}$ can be recognized by CA but not by OCA in real time. In 1995 and 1996, Terrier [13,14] showed two languages which can be recognized by CA but not by OCA. One of the languages is $\{uvv: u, v \in \{0, 1\}^*, |u| \geq 1\}$. The other is the language $L_0L_0$, where

$$L_0 = \{w: w = 1^j0^j \text{ or } w = 1^j010^j, y \in \{0, 1\}^*, j > 0\}.$$ 

Note that $L_0L_0$ is a context free language. The existence of this language implies that $\mathcal{C}$ (real-time OCAL) is not closed under concatenation, since $L_0$ is a real-time OCAL. Nakamura [9] showed a pumping lemma for recognition of cyclic strings by OCAs.

Concerning the limitation of real-time recognition of CA, Ibarra and Jiang [6] showed the following propositions in 1988:

- $\mathcal{C}$ (real-time CAL) is closed under reversal, if and only if $\mathcal{C}$ (real-time CAL) = $\mathcal{C}$ (linear-time CAL); and
- if $\mathcal{C}$ (real-time CAL) is closed under reversal, then it is also closed under concatenation.

Recently, Terrier [15] showed that if $\mathcal{C}$ (real-time CAL) is closed under concatenation, real-time recognition of CAL is as powerful as the linear-time recognition on unary languages.

1.3. Subjects of this paper

In this paper, we investigate language recognition of the strings containing binary representations of their own lengths to clarify limitations on the parallel language recognition power of CA. We denote the binary representation of the number $N$ by $B(N)$. We show that:

1. the language $L_Z = \{w \in \{0, 1\}^+: B(|w|)$ is a postfix of $w\}$ and its reversal $L_1^R$ are recognizable by CA in real time, but not recognizable by OCAL in real time;
2. the language $L_X = \{w \in \{0, 1\}^+: w$ contains $B(|w|)\}$ is a linear-time CAL, but not a real-time CAL; and
3. the language $L_Z[0, 1]$ is also not a real-time CAL. This implies that the class of real-time CALs is not closed under concatenation and under reversal.

The second result is a negative solution to the problem posed by Smith III [10], “Is the real-time recognition power of CA equivalent to the linear-time recognition power?” The third result is also a solution to the problem posed by Smith III [10]. These two problems have remained unsolved since he first posed them in 1972.

The difference in the power of real-time and linear time recognition is related to the use of the length information presented. In linear-time recognition, a CA can determine acceptance or rejection after receiving the length information. On the other hand, in real-time recognition, no CA can sufficiently make use of this information, as the rightmost cell determines acceptance or rejection at the same time it receives the signal from the leftmost cell.

2. Preliminaries

The following facts flow directly from the definitions above and/or from well-known CA studies [3,10].

1. Real-time recognition in time $|w| - 1$ is equivalent to that in time $|w|$, for any string $w$. This is important since the $|w| - 1$ recognition does not require the left boundary symbol.
2. For any real-time CAL $L$ over $\Sigma$, $\Sigma^*L$ is a real-time CAL.
3. For any real-time OCAL $L$ over $\Sigma$, $L\Sigma^*$ is also a real-time OCAL.
We use the following notations, which are well known in computational complexity theory. For functions \( f, g : N \to N \), where \( N \) is the set of positive integers, with \( f(n) > 0 \) and \( g(n) > 0 \), we write:

- \( f = O(g) \), if there exist constants \( n_0 \) and \( c \) such that \( f(n) \leq cg(n) \) for all \( n \geq n_0 \);
- \( f = \Theta(g) \), if \( f = O(g) \) and \( g = O(f) \); and
- \( f = \omega(g) \), if \( \lim_{n \to \infty} g(n)/f(n) = 0 \). (\( f \) grows faster than \( g \)).

3. Languages recognizable by CA in real time

In this section, we discuss real-time recognition of the language

\[ L_Z = \{ uv : u, v \in \{0, 1\}^*, v = B(|uv|) \}, \]

where \( B(|x|) \) is the binary number representing the length of string \( x \). Some example strings in \( L_Z \) are 1, 10, 011, 111, 0100, 1100, 00101, 01101, ..., where the underlines represent the binary representations of the lengths. Recognition of this language is closely related to that of the language \( L_X \) discussed in the next section. The author has shown the proof of the following proposition in [9].

**Proposition 1.** The language \( L_Z \) and its reversal \( L_Z^R \) are not real-time OCALs, but real-time CALs.

The former part of the proposition is proved based on the pumping lemma for cyclic strings in [9]. The language \( L_Z \) is an example of real-time CALs that are closed under reversal.

Since the latter part of this proposition is closely related to Theorem 3, we outline the construction of a CA which recognizes \( L_Z \) in [9]. The CA simulates binary counters; each cell has two modulo-2 counters to enumerate the lengths of the strings. It has the following two layers (Fig. 2).

1. In the first layer, the signals representing 0s and 1s in the input string move to the right until the signals meet the control signal, \( C_0 \), moving from the rightmost cell to the left. Then, these signals remain in the cell.
2. In the second layer, each cell contains the counters which generate binary numbers in numeric order on the diagonal which starts from the rightmost cell in the space–time diagram as shown in the right table in Fig. 2. The generated binary numbers are represented by dotted lines in the left diagram in Fig. 2. The cells containing two bits of the binary numbers also check whether the binary digits in the generated numbers are equal to the remaining signals in the first layer.

The result of the test is propagated by signal \( T \) in Fig. 2. If all the digits of the generated numbers coincide with the signals representing \( B(|uv|) \), signal \( E \) is generated. This signal moves to the left and reflects to the right when it meets control signal \( C_1 \) at velocity \( 1/3 \). When this signal arrives at the rightmost cell, it has an accepting state.

![Fig. 2. Transition in CA recognizing \( L_1 \) and generation of binary numbers (right).](image_url)
4. Languages not recognizable by CA in real time

In this section, we discuss real-time recognition of the language

\[ L_X = \{uvw: u, v, w \in \{0, 1\}^n, v = B(\lfloor uvw \rfloor)\}, \]

which is a set of strings containing the binary representations of their own lengths. Some example strings in \( L_X \) are

1, 10, 011, 110, 0100, 1100, 1000, 1001, 01010, 010110, \ldots.

**Theorem 2.** The language \( L_X \) is not a real-time CAL.

The proof of this theorem is based on the following observation. Suppose that a CA \( S \) recognizes \( L_X \). Then, for any string \( x \in \{0, 1\}^n \) and any integer \( N \) with \( 2 \leq N \leq n \), if there are strings \( u, v, w \) with \( x = uvw \), \( v = B(N) \), and \( |vw| \leq N - 1 \), then the rightmost cell in \( S \) has an accepting state at time \( N - 1 \). The transition of configurations in \( S \) needs to have a means for independent “information propagation,” or “channels,” from the places of the binary representations to the rightmost cell. Figure 3 represents the hypothetical channels as broken lines. For each symbol \( b \) in \( x \in \{0, 1\}^n \), \( n = 2^k \), if we consider only the binary representations of length \( k \), then \( k \) binary numbers include this symbol, and the number of the channels from \( b \) is generally of the magnitude \( \Theta(\log n) \). In fact, the number of channels from \( b \) grows faster than \( \log n \), since there are correspondences between substrings that are longer or shorter than \( \log_2 n \) and the rightmost cell.

Note that the signals in the channels need to have some information about the directions of propagation, which we ignore here. The properties of the possible signals in CA are investigated in [8].

For any configuration \( u = s_ns_{n-1} \cdots s_1 \# \), let \( [u]_l \) denote the \( i \)th state \( s_i \). The state of the rightmost cell in the configuration \( c \) is denoted by \([c]_l\).

**Proof of Theorem 2.** Suppose that there is a CA \( S = (K, \#, f, A) \) which recognizes \( L_X \). We consider the transition of any initial configuration \( x\# \) in \( S \) with \( x \in \{0, 1\}^n \), \( n = 2^k \), for any integer \( k \geq 1 \). Let \( x_p \in \{0, 1\}^n \) denote a string such that all symbols except the \( p \)th symbol in \( x_p \) are identical to those in \( x \). We call a sequence \( p = p_0, p_1, \ldots, p_l \) of integers a channel from \( p \), if \( |p_{i-1} - p_i| \leq 1 \) and \( [f^i(x_p)]_{p_i} \neq [f^i(x_p)]_{p_{i-1}} \) for all \( i \in \{1, 2, 3, \ldots, l\} \).

For any \( p \) with \( k \leq p \leq n - k \), there are at most \( k \) integers \( N \), such that the \( k \) consecutive symbols containing the \( p \)th symbol in \( x \) represent \( N \). For each of the binary numbers \( B(N) \), if \( N \leq p \) then \( [f^{N-1}(x)]_{l} \in A \), i.e. the rightmost cell has an accepting state at time \( N - 1 \). If \( x \) contains exactly one binary number \( B(N) \), then \( [f^{N-1}(x)]_{l} \neq A \), and the channel is of length \( N - 1 \). If the string \( x \) contains another occurrence of \( B(N) \), the channel may connects \( p \) to any point of the other hidden channel and its length is shorter than \( N - 1 \). However, the channel length is of magnitude of \( n, \Theta(n) \), in both cases. As the CA needs to recognize strings in \( L_X \) of unrestricted lengths, there are channels for binary representations of lengths that are shorter or longer than \( k \). Therefore, the number of channels starting with \( p \) is \( k = \omega(\log n) \), i.e., the number grows faster than \( \log n \).

![Fig. 3. An illustration of the transition of CA S in the proof of Theorem 2.](image-url)
Next, consider the transition for initial configurations in which two or more symbols are different from those in $x$. Let $x_{pq} \in \{0, 1\}^n$ be a string such that all symbols except the $p$th and $q$th symbols are identical to those in $x$ for any $q$ with $k \leq q < p \leq n - k$. If $p - q > k$, although the channels from $p$ may be changed by the value of $q$th symbol, the number and lengths of channels are independent of the value. This condition needs to be satisfied for a string $x_{q_1 \ldots q_m}$, such that all symbols are identical to those in $x$ except in the positions $q_1, q_2, \ldots, q_m$ with $n - k \geq q_1, q_m \geq k$, $q_{i+1} - q_i > k$ for all $i \in \{1, 2, \ldots, m - 1\}$ and $m = \Theta(n/\log n)$. Therefore, there is a string $x \in \{0, 1\}^n$ such that the number of possible channels in $x_{q_1 \ldots q_m}$ is

$$\omega \left( \frac{n}{\log n} \log n \right) = \omega(n)$$

and the channels are of lengths $\Theta(n)$. This implies that there is a cell such that the number of channels including this cell cannot be restricted to a constant, when $n$ is sufficiently large. On the other hand, if $j$ channels include a common cell at a time for any $j \geq 2$, then the cell needs to have at least $2^j$ states. This, however, contradicts the assumption that the state set $K$ is finite. Therefore, there does not exit a CA that recognizes $L_X$.

**Theorem 3.** The language $L_X = \{uvw : u, v, w \in \{0, 1\}^*, v = B(|uvw|)\}$ is a linear-time CAL.

**Proof.** In linear-time recognition, there is a CA in which all cells have the firing state at time $|uvw|$ by the firing squad synchronization from both the leftmost and rightmost cells. Therefore, we can construct a CA which generates binary numbers in the right cells, holds the binary number $B(|uvw|)$ after the synchronization, and tests whether the input string contains the binary number as shown in Fig. 4. Signals $T$ and $E$ play roles similar to those in the CA in Fig. 2. When the rightmost cell receives signal $E$, it has an accepting state. □

5. Closure properties on the class of real-time CALs

**Theorem 4.** $C$(real-time CAL) is not closed under concatenation.
Proof. We show that the following set, which is the concatenation of two real-time CALs $L_Z$ and $\{0, 1\}^+$, is not a real-time CAL,

$$L_Z \{0, 1\}^+ = \{ uvw : u, v \in \{0, 1\}^*, w \in \{0, 1\}^+, v = B(|uv|) \}.$$  

Suppose that there is a CA $S = (K, \#, f, A)$ which recognizes this language in real time. For any initial configuration $x\#, x \in \{0, 1\}^n$ and an integer $N$, if $B(N)w$ is a postscript of $x$ and $n > N + |w|$, then $S$ has an accepting state at time $N + |w| - 1$. We omit the remainder of the proof which reveals that (1) there is a string $x \in \{0, 1\}^n$ such that the number of possible channels is $\omega(n)$ and the channels are of lengths $\Theta(n)$; and (2) this contradicts that the state set is finite, because the proof is similar to that of Theorem 2. □

Theorem 5. $C($real-time CAL$)$ is not closed under reversal.

This theorem is a consequence of Theorem 4, since Ibarra and Jiang [6] showed that if the class of real-time CALs is closed under reversal, then it is also closed under concatenation. We show a direct proof as follows.

Proof of Theorem 5. Language $L_Z \{0, 1\}^+$ is not a real-time CAL by Theorem 4. On the other hand, the reversal of this language, $(L_Z \{0, 1\}^*)^{-R} = \{0, 1\}^+L_Z^{-R}$ is a real-time CAL, since $L_Z^{-R}$ is a real-time CAL by Proposition 1. □

Note that this theorem can also be proved apart from Theorem 4. Ibarra and Jiang showed that $C($real-time CAL$)$ is closed under reversal if and only if $C($linear-time CAL$) = C($real-time CAL$)$, where the equality of the two classes is refuted by Theorem 2.

6. Conclusion

In this paper, we discussed language recognition of strings containing binary representations of their lengths by CA and proved the following propositions.

1) There is a language $(L_X)$ not recognizable in real time by CA but recognizable in linear time. Hence, $C($real-time CAL$) \subset C($linear-time CAL$)$.

2) The class of real-time CA languages is not closed under concatenation, and the class is not closed under reversal. Hence, $C($real-time (right) CAL$) \neq C($real-time left CAL$)$.

Fig. 5. Relations between the language classes.
These results are solutions to the problems posed by Smith III [10]. Figure 5, which is an updated version of the diagram in [6], summarizes the relations among the language classes. The arrows represent proper inclusion relations unless otherwise specified. The double boxes represent the classes that are proved to be closed under concatenation and under reversal, and single boxes those not closed under concatenation. The broken line enclosing the box of linear-time CAL indicates that the closure problem under concatenation of this class is open.

There remain fundamental open problems related to real-time and linear-time CALs as follows [10].

- What are other languages not recognizable in real-time by CA? Is the language \( \{ww : w \in \{0, 1\}^+\}\{0, 1\}^* \) a real-time CAL?
- Is there any context-free language that is not a real-time CAL?
- Is the class of linear-time CALs closed under concatenation?
- What is a non-linear-time CA language?
- Is the class of CALs equivalent to that of OCALs?

The last two important problems are closely related to each other, since Ibarra and Jiang [6] showed that if CA is more powerful than OCA, then non-linear-time CA is more powerful than linear-time CA.

Acknowledgment

The author would like to thank Professor Véronique Terrier for her comments, which were helpful in improving the proof of Theorem 2.

References