The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Some state-efficient algorithms for real-time generation of non-regular sequences on cellular automata

Naoki Kamikawa and Hiroshi Umeo

Graduate School of Engineering, Osaka Electro-Communication University
Neyagawa-shi, Hatsu-cho 18-8, Osaka, 572-853, Japan
(*Tel:* +81-72-824-1131, *E-mail:* naoki@jtw.zaq.ne.jp, umeo@cyt.osakac.ac.jp)

Abstract **:** A model of cellular automata (CA) is considered to be a non-linear model of complex systems in which an infinite one-dimensional array of finite state machines (cells) updates itself in a synchronous manner according to a uniform local rule. We study a sequence generation problem on the CA with small number of states and propose several state-efficient real-time sequence generation algorithms for non-regular sequences. We show that a sequence $\{2^n \mid n = 1, 2, 3, \ldots\}$ can be generated in real-time by a CA with 3 states and a sequence $\{n^2 \mid n = 1, 2, 3, \ldots\}$ can be generated in real-time by a CA with 4 states. We also study infinite non-regular sequences generated on CA with 2 states.

Keywords **:** cellular automata, real-time sequence generation problem

1 Introduction

A model of cellular automata (CA) was devised originally for studying self-reproduction by John von Neumann. It is now studied in many fields such as complex systems. We study a sequence generation problem on the CA. Arisawa[1], Fischer[2] and Korec[3] studied generation of a class of natural numbers on CA. In this paper, we show that a sequence $\{2^n \mid n = 1, 2, 3, \ldots\}$ can be generated in real-time by a CA with 3 states and a sequence $\{n^2 \mid n = 1, 2, 3, \ldots\}$ can be generated in real-time by a CA with 4 states. We also study infinite non-regular sequences generated on the CA with 2 states.

2 Real-time sequence generation problem on CA

A cellular automaton consists of an infinite array of identical finite state automata, each located at a positive integer point (See Fig. 1).

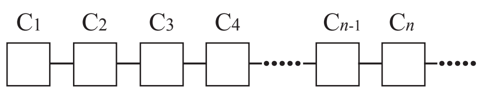

Figure 1: Cellular automaton.

Each automaton is referred to as a cell. A cell at point i is denoted by C_i, where $i \geq 1$. Each C_i, except for C_1, is connected to its left- and right-neighbor cells via a communication link. Each cell can know state of its left- and right-neighbor cells via communication link. One distinguished leftmost cell C_1, the communication cell, is connected to the outside world. A cellular automaton (abbreviated by CA) consists of an infinite array of finite state automata $A = (Q, \delta, F)$, where

1. Q is a finite set of internal states.
2. δ is a function defining the next state of a cell, such that δ: $Q \times Q \times Q \rightarrow Q$, where $\delta(\mathtt{p}, \mathtt{q}, \mathtt{r}) = \mathtt{s}$, $\mathtt{p}$, $\mathtt{q}$, $\mathtt{r}$, $\mathtt{s} \in Q$ has the following meaning: We assume that at step t the cell C_i is in state $\mathtt{q}$, the left cell C_{i-1} is in state $\mathtt{p}$ and the right cell C_{i+1} is in state $\mathtt{r}$. Then, at the next step $t+1$, C_i assumes state $\mathtt{s}$. The leftmost cell C_1 is connected to the outside world. The outside world is expressed by $*$. A quiescent state $\mathtt{q} \in Q$ has a property such that $\delta(\mathtt{q}, \mathtt{q}, \mathtt{q}) = \mathtt{q}$.
3. $F \subseteq Q$ is a special subset of Q. The set F is used to specify a designated state of C_1 in the definition of sequence generation.

We now define the **sequence generation problem** on CA. Let M be a CA and $\{t_n \mid n = 1, 2, 3, ...\}$ be an infinite monotonically increasing positive integer sequence defined natural numbers, such that $t_n \geq n$ for any $n \geq 1$. We then have a semi-infinite array of cells, as shown in Fig. 1, and all cells, except for C_1, are in the quiescent state at time $t = 0$. The communication cell C_1 assumes a special state $\mathtt{r}$ in Q for initiation of the sequence generator. We say that M generates a sequence $\{t_n \mid n = 1, 2, 3, ...\}$ in k *linear-time* if and only if the leftmost end cell of M falls into a special state in $F \subseteq Q$ at time $t = k \cdot t_n$, where k is a positive integer. We call M a *real-time* generator when $k = 1$.

3 Real-time generation of non-regular sequences

Arisawa[1], Fischer[2] and Korec[3] studied real-time generation of non-regular sequences on CA. Arisawa[1] shows that sequence $\{2^n \,|\, n = 1,2,3,\ldots\}$ can be generated in 2 linear-time by a CA with 7 states and sequence $\{n^2 \,|\, n = 1,2,3,\ldots\}$ can be generated in 2 linear-time by a CA with 9 states. Korec[3] shows that prime sequences can be generated in real-time by a CA with 9 states. In this paper, we show that sequence $\{2^n \,|\, n = 1,2,3,\ldots\}$ can be generated in real-time by a CA with 3 states and sequence $\{n^2 \,|\, n = 1,2,3,\ldots\}$ can be generated in real-time by a CA with 4 states.

3.1 Sequence $\{2^n \,|\, n = 1,2,3,\ldots\}$

Sequence $\{2^n \,|\, n = 1,2,3,\ldots\}$ can be generated in real-time by a CA with 3 states that is given in Table 1. In Fig. 2, we show a time-space diagram for real-time generation of sequence $\{2^n \,|\, n = 1,2,3,\ldots\}$.

Q	Right State: Q	Ac	N	*
Left State: Q	Q		N	Q
Ac	Q		N	
N	N		Q	
*	Q		Ac	

Ac	Right State: Q	Ac	N	*
Left State: Q				
Ac				
N				
*	Q			

N	Right State: Q	Ac	N	*
Left State: Q	Q			
Ac				
N				
*	Q			

Table 1: Transition rules for real-time generation of sequence $\{2^n \,|\, n = 1,2,3,\ldots\}$.

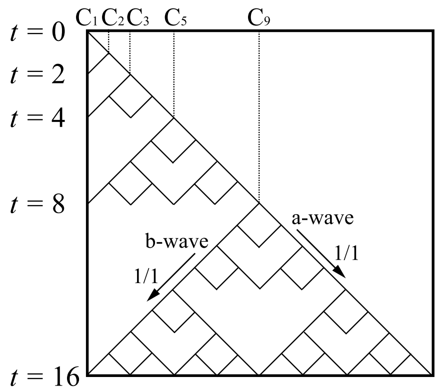

Figure 2: Time-space diagram for real-time generation of sequence $\{2^n \,|\, n = 1,2,3,\ldots\}$.

Let i be any positive integer such that $i \geq 2$. When cell C_i is in state `Q` and C_i's left- or right-neighbor cell is in state `N`, $\delta(\texttt{Q},\texttt{N},\texttt{Q}) = \texttt{N}$ or $\delta(\texttt{Q},\texttt{Q},\texttt{N}) = \texttt{N}$ are applied in C_i, then a state of C_i changes to `N`. C_i is in state `N` and C_i's left- and right-neighbor cells is in state `Q`, $\delta(\texttt{N},\texttt{Q},\texttt{Q}) = \texttt{Q}$ is applied in C_i, then a state of C_i changes to `Q`. As a result, the state `N` advances toward the left or right at at speed 1-cell/1-step in cell space. The state `N` which propagates right is called *a-wave* and the state `N` which propagates left is called *b-wave*. The a-wave generates the b-wave every one step. The b-wave generates the a-wave every one step. When the a-wave collides with the b-wave, the a- and b-waves are deleted. Because, the a-wave collides with the b-wave, namely C_i is in state `Q` and C_i's left- and right-neighbor cells is in state `N`, $\delta(\texttt{Q},\texttt{N},\texttt{N}) = \texttt{Q}$ is applied in C_i, then a state of C_i changes to `Q`. We define $\texttt{Ac} \in F$. When the b-wave reaches the leftmost cell C_1, namely C_1 is in state `Q` and C_1's right-neighbor cell is in state `N`, $\delta(\texttt{Q},*,\texttt{N}) = \texttt{Ac}$ is applied in C_1, then a state of C_1 changes to `Ac`.

Let x, n and k be any natural number. Let y be any positive integer. At time $t = y$, cell C_x takes state `N` and other cells takes a state `Q` (See Fig. 3).

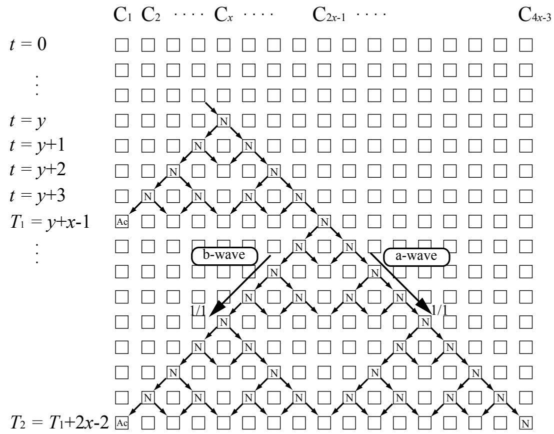

Figure 3: Time-space diagram for real-time generation of sequence $\{2^n \,|\, n = 1,2,3,\ldots\}$.

The a- and b-waves are generated on cell C_x. When the b-wave reaches C_1, the cell C_1 takes state `Ac`. Time when C_1 takes the state `Ac` is assumed to be T_n. Because the b-wave's speed is 1/1, it is approved that $T_1 = y + x - 1$. Next, the b-wave reached to C_1 is generated on C_{2x-1} at time $t = T_1$. Because the a-wave is not generated on C_1 at time $t = T_1$ and the b-wave generated on C_x exists in C_{2x-1} at time $t = T_1$. Therefore, it is approved that $T_2 = T_1 + 2(x-1)$, $T_3 = T_2 + 4(x-1)$, $T_4 = T_3 + 8(x-1)$, $T_5 = T_4 + 16(x-1)$,$\ldots$,$T_n = T_{n-1} + 2(x-1)\cdot 2^{k-1}$. T_n's difference sequence is assumed to be b_k such that $b_k = 2(x-1)\cdot 2^{k-1}$. Therefore, it is approved that $T_n = T_1 + \sum_{k=1}^{n-1} b_k = y + x - 1 + \frac{2(x-1)(2^{n-1}-1)}{2-1} = 2(x-1)(2^{n-1}-1) + x + y - 1$. We assume that the initial configuration is the leftmost cell C_1 takes state `N` and other cell take a state `Q`. At time $t = 1$, cell C_1 takes state `N` and other cell take a state `Q`. Therefore, it is approved that $x = 2, y = 1$ and $T_n = 2(x-1)(2^{n-1}-1) + x + y - 1 = 2^n$. C_1 takes state `Ac` at time $t = 2^n$ $(n = 1,2,3,\ldots)$. It is seen that the scheme given above can exactly generate sequence $\{2^n \,|\, n = 1, 2, 3, \ldots\}$ in real-time. We have implemented the algorithm on a computer. We have tested the validity of the rule set from $t = 0$ to $t = 20000$ steps. We obtain the following theorem.

[Theorem 1] Sequence $\{2^n \,|\, n = 1, 2, 3, \ldots\}$ can be generated by a CA with 3 states in real-time.

In Fig. 4, we show a number of snapshots of the configuration from $t = 0$ to 16.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	N	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
1	Q	N	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
2	Ac	Q	N	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
3	Q	N	Q	N	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
4	Ac	Q	Q	Q	N	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
5	Q	Q	Q	N	Q	N	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
6	Q	Q	N	Q	Q	Q	N	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
7	Q	N	Q	N	Q	N	Q	N	Q	Q	Q	Q	Q	Q	Q	Q	Q
8	Ac	Q	Q	Q	Q	Q	Q	Q	N	Q	Q	Q	Q	Q	Q	Q	Q
9	Q	Q	Q	Q	Q	Q	Q	N	Q	N	Q	Q	Q	Q	Q	Q	Q
10	Q	Q	Q	Q	Q	Q	N	Q	Q	Q	N	Q	Q	Q	Q	Q	Q
11	Q	Q	Q	Q	Q	N	Q	N	Q	N	Q	N	Q	Q	Q	Q	Q
12	Q	Q	Q	Q	N	Q	Q	Q	Q	Q	Q	Q	N	Q	Q	Q	Q
13	Q	Q	Q	N	Q	N	Q	Q	Q	Q	Q	N	Q	N	Q	Q	Q
14	Q	Q	N	Q	Q	Q	N	Q	Q	Q	N	Q	Q	Q	N	Q	Q
15	Q	N	Q	N	Q	N	Q	N	Q	N	Q	N	Q	N	Q	N	Q
16	Ac	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	N

Figure 4: A configuration of real-time generation of sequence $\{2^n \,|\, n = 1, 2, 3, \ldots\}$.

3.2 Sequence $\{n^2 \,|\, n = 1, 2, 3, \ldots\}$

Sequence $\{n^2 \,|\, n = 1, 2, 3, \ldots\}$ can be generated in real-time by a CA with 4 states that is given in Table 2. In Fig. 5, we show a time-space diagram for real-time generation of sequence $\{n^2 \,|\, n = 1, 2, 3, \ldots\}$.

Q		Right State: Q	N	Ac	R	*
Left State	Q	Q				Q
	N	Q				
	Ac	N				
	R	Q				
	*		N	Q	Q	

N		Right State: Q	N	Ac	R	*
Left State	Q	R			R	
	N					
	Ac					
	R	R			R	
	*	Ac			Ac	

Ac		Right State: Q	N	Ac	R	*
Left State	Q	R			R	
	N					
	Ac					
	R	R			R	
	*	Q			Q	

R		Right State: Q	N	Ac	R	*
Left State	Q		N	R	R	
	N	R			R	
	Ac	Ac			Ac	
	R	R	N	R	R	
	*					

Table 2: Transition rules for real-time generation of sequence $\{n^2 \,|\, n = 1, 2, 3, \ldots\}$.

Real-time generation of sequence $\{n^2 \,|\, n = 1, 2, 3, \ldots\}$ is described in terms of two waves: *a-wave* and *b-wave*. Let i be any positive integer such that $i \geq 2$. When cell C_i is in state R, C_{i-1} is in state Ac and C_{i+1} is in state Q or R, $\delta(\mathtt{R}, \mathtt{Ac}, \mathtt{Q}) = \mathtt{Ac}$ or $\delta(\mathtt{R}, \mathtt{Ac}, \mathtt{R}) = \mathtt{Ac}$ are applied in C_i, then a state of C_i changes to Ac. The state Ac which propagates right is called a-wave. When cell C_i is in state R, C_{i-1} is in state Q or R and C_{i+1} is in state N, $\delta(\mathtt{R}, \mathtt{R}, \mathtt{N}) = \mathtt{N}$ or $\delta(\mathtt{R}, \mathtt{Q}, \mathtt{N}) = \mathtt{N}$ are applied in C_i, then a state of C_i changes to N. The state N which propagates left is called b-wave. We assumes that $\mathtt{Ac} \in F$. Let i, k and l be any positive integer and let j be any positive natural number. At time $t = j$, it is assumed that the

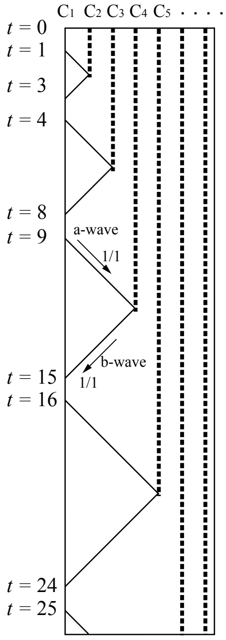

Figure 5: Time-space diagram for real-time generation of sequence $\{n^2 \,|\, n = 1, 2, 3, \ldots\}$.

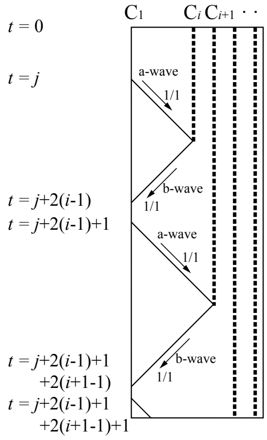

Figure 6: Time-space diagram for real-time generation of sequence $\{n^2 \,|\, n = 1, 2, 3, \ldots\}$.

leftmost cell C_1 takes state Ac, cell $C_k (2 \leq k \leq i - 1)$ take a state R and cell $C_l (l \geq i)$ take a state Q (See Fig. 6). The a-wave, generated by C_1 at time $t = j$, propagates in the right direction at $1/1$ speed. The a-wave reaches C_i at time $t = j + i - 1$. A state of cell C_i changes to N. The a-wave reflects on C_i and the b-wave is generated. The b-wave propagates in the left direction at $1/1$ speed. The b-wave reaches C_1 at time $t = j + 2(i - 1)$. A state of C_1 changes to N. At the next step $t = j + 2(i - 1) + 1 = j + 2i - 1$, a state of cell C_1 changes to Ac. Next, it is $t = j + 2i + 1 + 2i + 1 = 4i + 2$ that C_1 takes state Ac. Because, the a-wave reflects on C_{i+1} by the state of C_{i+1} changed to R. Therefore, C_1 takes state Ac at time $t = j + n^2 + 2(i - 2)n - 2i + 3 \, (n = 1, 2, 3, \ldots)$. Let m be any positive integer. When the initial configuration is the leftmost cell C_1 takes state N and cell $C_m (m \geq 2)$ take a state Q, it is approved that $j = 1, i = 2$. Therefore, C_1 takes state Q at time $t = n^2 \, (n = 1, 2, 3, \ldots)$. It is seen that the scheme given above can exactly generate sequence $\{n^2 | n = 1, 2, 3, \ldots\}$ in real-time. We have implemented the algorithm on a computer. We have tested the validity of the rule set from $t = 0$ to $t = 20000$ steps. We obtain the following theorem.

[Theorem 2] Sequence $\{n^2 \,|\, n = 1, 2, 3, \ldots\}$ can be generated by a CA with 4 states in real-time.

In Fig. 7, we show a number of snapshots of the configuration from $t = 0$ to 25.

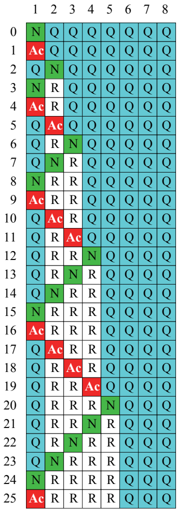

Figure 7: A configuration of real-time generation of sequence $\{n^2 \,|\, n = 1, 2, 3, \ldots\}$.

4 Sequences which can be generated on CA with 2 states

In this section, we study sequences which can be generated on CA with 2 states. We show sequences which can be generated on CA with 2 states by using a personal computer. Let A be a CA with 2 state and i, k be any natural number. A consists of an infinite array of finite state automata $A = (Q, \delta, F)$, where $Q = \{\mathtt{Q}, \mathtt{N}\}$, $F = \{\mathtt{N}\}$. The initial configuration is the leftmost cell C_1 takes state N and cell $C_k (k \geq 2)$ take a quiescent state Q . The cell C_i can take 2 state such that Q and N. The right C_{i+1} can take 2 state such that Q and N. The left C_{i-1} can take 3 state such that Q, N and $*$. Because, the leftmost cell C_1 is connected to the outside world. Moreover, the function δ is defined by δ: $Q \times Q \times Q \rightarrow Q$ and a quiescent state $\mathtt{Q} \in Q$ has a property such that $\delta(\mathtt{Q}, \mathtt{Q}, \mathtt{Q}) = \mathtt{Q}$. Therefore, there are $2^{\{(2 \cdot 2 \cdot 3) - 1\}} = 2048$ transition rules. It simulates by 2048 transition rules with a personal computer and the generated sequences are examined. Table 3 shows the class of sequences generated on CA with 2 states.

Class	Example
non-regular sequences	$\{2^{n+1} - 2 \,|\, n = 1\ 2\ 3 \quad \}$
arithmetical sequences	$\{4n - 2 \,|\, n = 1\ 2\ 3 \quad \}$
finite sequences	$\{2\ 4\}$
union of more sequences	$\{2\} \cup \{2^{n+1} + 1 \,|\, n = 1\ 2\ 3 \quad \} \cup$ $\{2^{n+2} - 1 \,|\, n = 1\ 2\ 3 \quad \}$
random sequences	-

Table 3: The class of sequences generated on CA with 2 states.

Fig. 8 shows generation of sequence $\{2\} \cup \{2^{n+1} + 1 \,|\, n = 1, 2, 3, \ldots\} \cup \{2^{n+2} - 1 \,|\, n = 1, 2, 3, \ldots\}$.

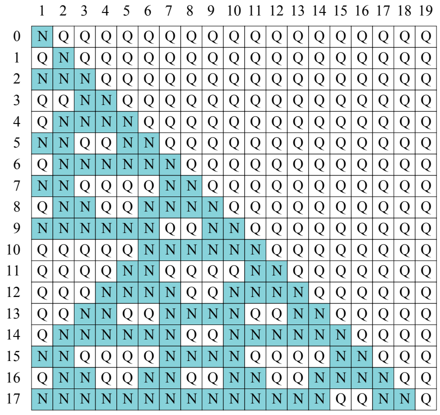

Figure 8: A configuration of real-time generation of sequence $\{2\} \cup \{2^{n+1} + 1 \,|\, n = 1, 2, 3, \ldots\} \cup \{2^{n+2} - 1 \,|\, n = 1, 2, 3, \ldots\}$.

The state transition does not happen on CA with 1 state. Therefore, sequence generation algorithms on CA with 2 states are lower bound. In addition, sequence $\{2^n \,|\, n = 1, 2, 3, \ldots\}$ can not be generated by a CA with 4 states in real-time. Therefore, the algorithm shown in section 3.1 is lower bound.

5 Conclusion

We study a sequence generation problem on CA. We showed that sequence $\{2^n \,|\, n = 1, 2, 3, \ldots\}$ can be generated in real-time by a CA with 3 states and $\{n^2 \,|\, n = 1, 2, 3, \ldots\}$ can be generated in real-time by a CA with 4 states. Several state-efficient real-time sequence generation algorithms for non-regular sequences have been proposed.

References

[1] M. Arisawa; On the generation of integer series by the one-dimensional iterative arrays of finite state machines (in Japanese), The Trans. of IECE, 71/8 Vol. 54-C, No.8, pp.759-766, (1971).

[2] P. C. Fischer; Generation of primes by a one-dimensional real-time iterative array. *J. of ACM*, Vol.12, No.3, pp.388-394, (1965).

[3] I. Korec; Real-time generation of primes by a one-dimensional cellular automaton with 9 states. *Proc. of MCU '98*, pp.101-116, (1998).

[4] N. Kamikawa and H. Umeo: Sequence Generation Problem on Communication-restricted Cellular Automata, The 5th WSEAS International Conference on Non-Linear Analysis, Non-Linear System and Chaos (NOLASC '06), pp.143-148, (2006).

[5] N. Kamikawa and H. Umeo; Real-Time Sequence Generation Problem on One-Bit Inter-Cell-Communication Cellular Automata,The 15th IEEE International Workshop on Nonlinear Dynamics of Electronic Systems (NDES 2007), pp.86-89, (2007).

[6] N. Kamikawa and H. Umeo; Some Algorithms for Real-Time Generation of Non-Regular Sequences on One-Bit Inter-Cell-Communication Cellular Automata, SICE Annual Conference 2007, pp.953-958, (2007).