# Sequence Generation Problem on Communication-restricted Cellular Automata

Naoki Kamikawa and Hiroshi Umeo
Univ. of Osaka Electro-Communication, Faculty of Information Science and Technology,
Neyagawa-shi, Hatsu-cho 18-8, Osaka, 572-8530, Japan

**Abstract:** Cellular automata (CA) are 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 a special restricted class of cellular automata having 1-bit inter-cell communications ($\mathrm{CA}_{1-\mathrm{bit}}$) and propose several state-efficient real-time sequence generation algorithms for non-regular sequences.

**Key words:** cellular automaton, sequence generation problem

## 1 Introduction

Cellular automata (CA) are 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 a special restricted class of cellular automata having 1-bit inter-cell communications ($\mathrm{CA}_{1-\mathrm{bit}}$) and propose several state-efficient real-time sequence generation algorithms for non-regular sequences. The 1-bit CA can be thought to be one of the most powerless and simplest models in a variety of CAs. First, in section 2, we introduce a cellular automaton with 1-bit inter-cell communication and define the sequence generation problem on $\mathrm{CA}_{1-bit}$. In section 3, it is shown that infinite non-regular sequences such as $\{2^n|\, n = 1, 2, 3, ..\}$ and Fibonacci sequences can be generated in real-time by cellular automata with 1-bit inter-cell communication. Those sequence generation algorithms will be realized on the 1-bit CAs with a relatively small number of internal states. It is also shown that an infinite prime sequence can be generated in real-time by a cellular automaton having 1-bit inter-cell communications ($\mathrm{CA}_{1-\mathrm{bit}}$). The algorithm presented is based on the classical sieve of Eratosthenes, and its implementation will be made on a $\mathrm{CA}_{1-\mathrm{bit}}$ using 34 internal states and 71 transition rules.

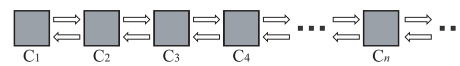


Figure 1: A one-dimensional cellular automaton with 1-bit inter-cell communication.

## 2 Sequence generation problem on $\mathrm{CA}_{1-\mathrm{bit}}$

A one-dimensional 1-bit inter-cell communication cellular automaton consists of an infinite array of identical finite state automata, each located at a positive integer point (See Fig. 1). Each automaton is referred to as a cell. A cell at point $i$ is denoted by $\mathrm{C}_i$, where $i \geq 1$. Each $\mathrm{C}_i$, except for $\mathrm{C}_1$, is connected to its left- and right-neighbor cells via a left or right one-way communication link. These communication links are indicated by right- and left-pointing arrows in Fig. 1, respectively. Each one-way communication link can transmit only one bit at each step in each direction. One distinguished leftmost cell $\mathrm{C}_1$, the communication cell, is connected to the outside world. A cellular automaton with 1-bit inter-cell communication (abbreviated by $\mathrm{CA}_{1-\mathrm{bit}}$) consists of an infinite array of finite state automata $A = (Q, \delta, F)$, where

1. $Q$ is a finite set of internal states.
2. $\delta$ is a function, defining the next state of any cell and its binary outputs to its left- and right-neighbor cells, such that $\delta$: $Q \times \{0,1\} \times \{0,1\} \rightarrow Q \times \{0,1\} \times \{0,1\}$, where $\delta(p,x,y) = (q,x',y')$, $p$, $q \in Q$, $x, x', y, y' \in \{0,1\}$, has the following meaning. We assume that at step $t$ the cell $\mathrm{C}_i$ is in state $p$ and is receiving binary inputs $x$ and $y$ from its left and right communication links, respectively. Then, at the next step, $t+1$, $\mathrm{C}_i$ assumes state $q$ and outputs $x'$ and $y'$ to its left and right communication links, respectively. Note that binary inputs to $\mathrm{C}_i$ at step $t$ are also outputs of $\mathrm{C}_{i-1}$ and $\mathrm{C}_{i+1}$ at step $t$. A quiescent state $q \in Q$ has a property such that $\delta(q,0,0) = (q,0,0)$.

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.

Thus, the $CA_{1-\mathrm{bit}}$ is a special subclass of *normal* (i.e., *conventional*) cellular automata. We now define the **sequence generation problem** on $CA_{1-\mathrm{bit}}$. Let $M$ be a $CA_{1-\mathrm{bit}}$, and let $\{t_n | n = 1, 2, 3, ...\}$ be an infinite monotonically increasing positive integer sequence defined for 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 $r$ in $Q$ and outputs 1 to its right communication link at time $t = 0$ for initiation of the sequence generator. We say that $M$ generates a sequence $\{t_n | n = 1, 2, 3, ...\}$ in $k$ *linear-time* if and only if the leftmost end cell of $M$ falls into a special state in $F \subseteq Q$ and outputs 1 via its leftmost communication link at time $t = kt_n$, 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[3] and Korec[4] studied real-time generation of a class of natural numbers on the conventional cellular automata, where $O(1)$ bits of information were allowed to be exchanged at one step between neighboring cells. In this section we propose several 1-bit communication cellular algorithms which generate non-regular infinite sequences in real-time. The first sequence we consider is $\{2^n | n = 1, 2, 3, ..\}$.

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

We show that the context-sensitive sequence $\{2^n | n = 1, 2, 3, ...\}$ can be generated in real-time by a 1-state $CA_{1-\mathrm{bit}}$. A transition rule set for the $CA_{1-\mathrm{bit}}$ $M$ generating the sequence is as follows: $M = \{Q, \delta\}$, where $Q = \{a, q\}$,

$$\delta(a,0,0) = (a,0,0), \quad \delta(a,0,1) = (a,1,0),$$
$$\delta(q,0,0) = (q,0,0), \quad \delta(q,0,1) = (q,1,1),$$
$$\delta(q,1,0) = (q,1,1), \quad \delta(q,1,1) = (q,0,0).$$

The leftmost cell $C_1$ always assumes a state $a$ and $C_i (i \geq 2)$ takes a state $q$ at any step. Figure 2 shows some snapshots for the real-time generation of the sequence. Small black right and left triangles ▶ and ◀, shown in Fig. 2, indicate a 1-bit signal transfer in the right or left direction between neighbour cells. A symbol in a cell shows its internal state.

**[Theorem 1]** An infinite sequence $\{2^n | n = 1, 2, 3, ..\}$ can be generated by a 1-state $CA_{1-\mathrm{bit}}$ in real-time.

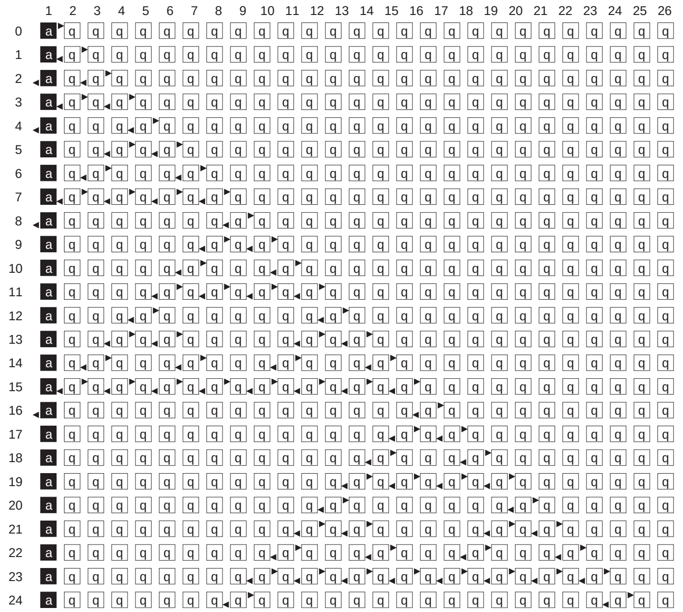

Figure 2: Snapshots for real-time generation of infinite sequence $\{2^n | n = 1, 2, ...\}$.

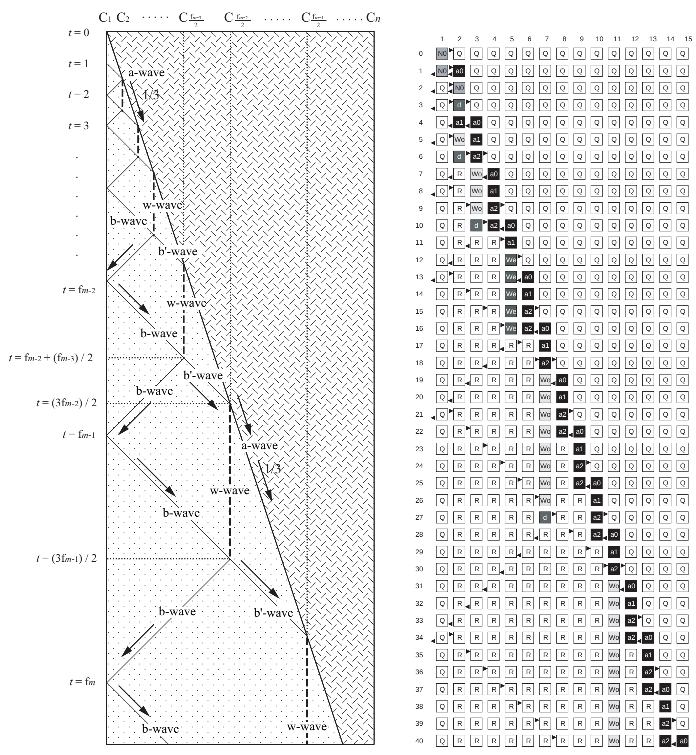


Fig. 3(a) Fig. 3(b)

Figure 3: Time-space diagram for real-time generation of Fibonacci sequence(Fig. 3(a)) and snapshot for its computer simulation(Fig. 3(b)).

| 1 | Q | R=0 | R=1 |
|---|---|---|---|
| | L=0 | (Q,0,0) | (Q,1,1) |
| | L=1 | (a0,1,0) | -- |

| 2 | R | R=0 | R=1 |
|---|---|---|---|
| | L=0 | (R,0,0) | (R,1,0) |
| | L=1 | (R,0,1) | -- |

| 3 | a0 | R=0 | R=1 |
|---|---|---|---|
| | L=0 | (a1,0,0) | -- |
| | L=1 | (N0,1,0) | -- |

| 4 | a1 | R=0 | R=1 |
|---|---|---|---|
| | L=0 | (a2,0,1) | (Wo,0,0) |
| | L=1 | (We,0,1) | -- |

| 5 | a2 | R=0 | R=1 |
|---|---|---|---|
| | L=0 | (a2,0,0) | (R,0,0) |
| | L=1 | (Wo,0,0) | (R,0,1) |

| 6 | We | R=0 | R=1 |
|---|---|---|---|
| | L=0 | (We,0,0) | (We,0,0) |
| | L=1 | (R,1,1) | -- |

| 7 | Wo | R=0 | R=1 |
|---|---|---|---|
| | L=0 | (Wo,0,0) | (Wo,0,0) |
| | L=1 | (d,0,1) | -- |

| 8 | N0 | R=0 | R=1 |
|---|---|---|---|
| | L=0 | (N0,1,1) | (Q,1,1) |
| | L=1 | (d,0,1) | -- |

| 9 | d | R=0 | R=1 |
|---|---|---|---|
| | L=0 | (R,1,0) | -- |
| | L=1 | (a1,1,0) | -- |

Table 1: Transition rules for real-time generation of Fibonacci sequence.

### 3.2 Fibonacci sequence

Next we consider Fibonacci sequence defined as $f_1 = 1$, $f_2 = 1$, $f_n = f_{n-1} + f_{n-2} (n \geq 3)$. Real-time generation of Fibonacci sequence is described in terms of four waves: *a-wave*, *b-wave*, *b'-wave* and *w-wave*. The a-wave, generated by $C_1$ at time $t = 0$, propagates in the right direction at 1/3 speed. The b-wave, generated on $C_1$ at time $t = 1$, oscillates between $C_1$ and w-wave and moves at 1/1 speed between them. The w-wave, which is generated on the intersecting point of a- and b'-waves, remains there till the next b-wave's arrival. It acts as a *wall*. The b'-wave is a split version of b-wave as will be described later. Whenever the b-wave arrives at $C_1$, $C_1$ takes a special state and outputs 1 to its left link. When the b-wave collides with w-wave from its left side, it is split into two b- and b'-waves. The former reflects there to the left and the latter proceeds in the right direction at 1/1 speed. Simultaneously, the w-wave vanishes. The split b'-wave generates a new w-wave when it meets a-wave and it itself disappears simultaneously. Fig. 3(a) is a time-space diagram showing the interactions of four waves given above.

Now we show straightforwardly how the Fibonacci sequence can be generated in real-time. The first two values of the Fibonacci sequence are such that $f_1 = 1$, $f_2 = 1$. We can construct the $CA_{1-bit}$ so that its left end cell outputs 1 at time $t = 1$. The output at time $t = 1$ is interpreted as two values given above. Let $m$ be any natural number such that $m \geq 3$. We assume that at time $t = f_{m-2}$, $C_1$ outputs 1 to its left link and the w-wave keeps its position on $C_{f_{m-3}/2}$. Then, we can get the following observations: The b-wave, generated by $C_1$ at time $t = f_{m-2}$, collides with the w-wave at time $t = f_{m-2} + (1/2) \cdot f_{m-3}$, and simultaneously it splits into b- and b'-waves. The b'-wave, proceeding in the right direction, collides with the a-wave at time $t = (3/2) \cdot f_{m-2}$ on $C_{f_{m-2}/2}$. The new w-wave, having disappeared on $C_{f_{m-3}/2}$, will be generated on $C_{f_{m-2}/2}$ at time $t = (3/2) \cdot f_{m-2}$. The b-wave, reflecting in the left direction, arrives at $C_1$ at time $t = f_{m-2} + f_{m-3} = f_{m-1}$ and outputs 1 to its left link. Afterwards the b-wave will reach $C_{f_{m-2}/2}$ at time $t = (3/2) \cdot f_{m-1}$ and it can always find the new w-wave, since $t = (3/2) \cdot f_{m-1} - (3/2) \cdot f_{m-2} = (3/2) \cdot f_{m-3} > 0$. It is easily seen by mathematical induction that the scheme given above can exactly generate the sequence in real-time. In Fig. 3(b) we show consecutive snapshots for the real-time generation of Fibonacci sequence on 1-bit CA with 9 internal states and 26 transition rules that is given in Table 1. Thus we have:

**[Theorem 2]** Fibonacci sequence can be generated by a $CA_{1-bit}$ in real-time.

### 3.3 Prime sequence

Arisawa [1], Fischer [3] and Korec [4] studied real-time generation of a class of natural numbers on the conventional cellular automata, where O(1) bits of information were allowed to be exchanged at one step between neighboring cells. Fischer [3] showed that prime sequences can be generated in real-time on the cellular automata with 11 states for $C_1$ and 37 states for $C_i (i \geq 2)$. Arisawa [1] also developed a real-time prime generator and decreased the number of states of each cell to 22. Korec [4] reported a real-time prime generator having 11 states on the same model. Umeo and Kamikawa [15] showed that the prime sequence can be generated in twice real-time by $CA_{1-\mathrm{bit}}$ having 54 internal states and 107 transition rules. In this section, we present a real-time prime generation algorithm on $CA_{1-\mathrm{bit}}$. The algorithm is implemented on a $CA_{1-\mathrm{bit}}$ using 34 internal states and 71 transition rules. Our prime generation algorithm is based on the well-known sieve of Eratosthenes. Imagine a list of all integers greater than 2. The first member, 2, becomes a prime and every second member of the list is crossed out. Then, the next member of the remainder of the list, 3, is a prime and every third member is crossed out. In Eratosthenes' sieve, the procedure continues with 5, 7, 11, and so on. In our procedure, given below, for any odd integer $k \geq 3$, every $2k$-th member of the list beginning with $k^2$ will be crossed out, since the $k$-th members less than $k^2$ (that is, $\{i \cdot k | 2 \leq i \leq k-1\}$) and $2k$-th members beginning with $k^2 + k$ (that is, it is an even number such that $\{(k + 2i - 1) \cdot k | i = 1, 2, 3, ...\}$) should have been crossed out in the previous stages. Thus, every $k$-th member beginning with $k^2$ is successfully crossed out in our procedure. Those integers never being crossed out are the primes. Figure 3 is a time-space diagram that shows a real-time detection of odd multiples of three, five and seven. In our detection, we use two 1-bit signals a- and b-waves, which will be described later, and pre-set partitions in which these two 1-bit signals bounce around.

We now outline the algorithm. Each cross-out operation is performed by $C_1$. We assume that the cellular space is initially divided by the partitions such that a special mark "$w$" is printed on cell $C_{i^2}$, for any positive integer $i \geq 1$. Those partitions will be used to generate reciprocating signals for the detection of odd multiples of, for example, three, five, and seven. We denote a subcellular space sandwiched by $C_k$ and $C_\ell$ as $S_i$, where $k = i^2$ ,

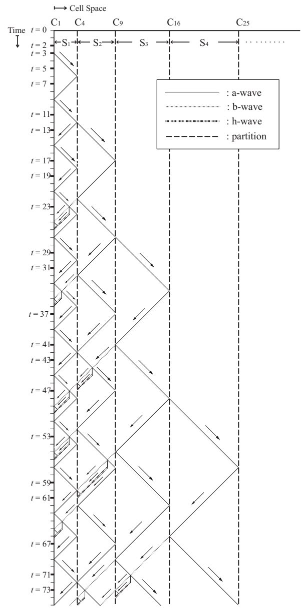


Figure 4: An h-wave that inhibits the reflection of 1-bit b-wave at the left end of each partition.

$\ell = (i+1)^2$ for any $i \geq 1$, and call it the $i$-th partition. Note that $S_i$ contains $(2i+2)$ cells, including both ends. The way to set up the partitions in terms of 1-bit communication will be described in Lemma 2.

**[Lemma 1]** Let $M$ be a $CA_{1-\text{bit}}$. We assume that the initial array of $M$ is partitioned into infinite blocks such that a special symbol "$w$" is marked on cells $C_{i^2}$, for any positive integer $i \geq 1$. The array $M$ given above can generate the $i$-th prime at time $t = i$.

**(Proof)** Consider a unit speed (1-cell/1-step) signal that reciprocates between the left and right ends of $S_i$, which is shown as the zigzag movements in Fig. 4. This signal is referred to as the *a-wave*. At every reciprocation, the a-wave generates a *b-wave* on the left end of $S_i$. The b-wave continues to move to $C_1$ at unit speed to the left through $S_{i-1}$, $S_{i-2}$, ..., $S_2$, and $S_1$. The b-wave generated at the left end of $S_i$ is responsible for notifying $C_1$ of odd multiples of odd integer $(2i+1)$ such that $(2j+1)(2i+1)$ for any positive integer $j \geq i$. In addition, the a-wave, on the second trip to the right end of $S_i$, initiates a new a-wave for $S_{i+1}$.

We assume that the initial a-wave for $S_i$ is generated on the left end of $S_i$ at time $t = 3i^2$. Then, as is shown in Fig. 4, the b-wave reaches $C_1$ at step $t = (2i+1)^2 + 2j \cdot (2i+1)$, where $j$ =0, 1, 2,.., . Moreover, the initiation of the first a-wave for $S_{i+1}$ is successfully made at step $t = 3(i+1)^2$ at the left end of $S_{i+1}$. We observe from Fig. 4 that the following signals are generated at the correct time. At time $t = 3$, the first a-wave starts toward the right from the cell $C_1$. At time $t = 6$, the a-wave arrives at the right end of $S_1$, and then is reflected toward the left and reaches $C_1$ at $t = 9$. The a-wave again proceeds toward the right at unit speed and reaches the right end of $S_1$. The first a-wave for $S_2$ is generated here. By mathematical induction, we see that the first a-wave for $S_i$ can be generated on the left end of $S_i$ at time $t = 3i^2$ for any $i \geq 1$. In this way, the b-wave generated at the left end of $S_i$ notifies $C_1$ of odd multiples of $(2i+1)$ that are greater than $(2i+1)^2$. Multiples of $(2i+1)$ that are less than $(2i+1)^2$ have been detected in the previous stages (See Fig. 4).

Whenever a left-traveling a-wave generated on $S_i$ and the left-traveling b-wave generated on $S_j$ $(j \geq i+1)$ start simultaneously at the right end of $S_i$, they are merged into one a-wave. Otherwise, the b-wave meets a reflected a-wave in $S_i$. Two kinds of unit speed left-traveling 1-bit signals exist in $S_i$ $(i \geq 1)$, that is, an a-wave that is reciprocating on $S_i$ and a b-wave that is generated on $S_j$ $(j \geq i+1)$. These two left-traveling 1-bit signals must be distinguished, since the latter does not produce a reciprocating a-wave. In order to avoid the reflection of the b-wave at the left end of each partition, we introduce a new *h-wave*, as shown in Fig. 4. Whenever a right-traveling a-wave and a left-traveling b-wave meet on a cell in $S_i$, the h-wave is newly generated at the next step on the cell in which they meet and, one step later, the h-wave begins to follow the left-traveling b-wave at unit speed. The h-wave stops the reflection of the b-wave at the left end of $S_i$, and then both waves disappear. A left-traveling a-wave and b-wave generated on $S_j$ $(j \geq i+1)$ always move with at least one cell interleaved between them. This enables the h-wave to be generated and transmitted toward the left. The left end cell $C_1$ has a counter that operates in modulo 2 and checks the parity of each step in order to detect every multiple of two. $C_1$ outputs a 1-bit signal to its left link if and only if it has not received any 1-bit signal from its right link at its previous step $t$. Then, $t$ is exactly prime.

In this way, the initially partitioned array given above can generate the prime sequence in real-time. In the next lemma, we show that the partition in the cellular space can be set up in time.

**[Lemma 2]** For any $i \geq 3$, the partition $S_i$ can be set up in time. Precisely, the right end cell $C_k$ of the $i$-th partition $S_i$, where $k = (i+1)^2$, can be marked at step $t = 3i^2 + 2i + 3$.

**(Proof)** For the purpose of setting up the partitions in the cellular space in time, we introduce seven new waves: *c-wave*, *d-wave*, *e-wave*, *f-wave*, *g-wave*, *u-wave* and *v-wave*. The direction in which these waves propagate and their propagation speeds are as follows:

| Wave | Direction | Speed |
|---|---|---|
| c-wave | right | 1/2 |
| d-wave | right | 1/1 |
| e-wave | right | 1/3 |
| f-wave | left | 1/1 |
| g-wave | right | 1/1 |
| u-wave | stays at a cell for only four steps and acts as a delay | 0 |
| v-wave | left | 1/1 |

The u-wave always stays at a cell for only four steps and acts as a delay for the generation of v-wave. Both operations for setting up the partitions and the generation and propagation of a- and b-waves described in the previous lemma are performed in parallel on the array. We make a small modification to the a-wave. The first reciprocation of the a-wave in each $S_i$ $(i \geq 3)$ is replaced by the c-, d-, e-, f-, u- and v-waves.

For any $i \geq 3$, we assume that:

- **$A_1$:** The c- and d-waves in $S_{i-1}$ arrive simultaneously at cell $C_k$, where $k = i^2$, and prints "$w$" as a right partition mark of $S_{i-1}$ on the cell at time $t = 3i^2 - 4i + 4$. The marking itself acts as a left partition of $S_i$.
- **$A_2$:** The g-wave in $S_{i-1}$ hits the right end of $S_{i-1}$ at time $t = 3i^2 - 2i$ and generates the c-wave in $S_i$.
- **$A_3$:** The a-wave in $S_{i-1}$ hits the right end of $S_{i-1}$ at time $t = 3i^2$ and generates the d- and e-waves for $S_i$ at time $t = 3i^2 + 2$.

Then, the following statements can be obtained.

- **$S_1$:** The c- and d-waves in $S_i$ arrive at the right end cell of $S_i$, $C_k$, where $k = (i+1)^2$, and the marker "$w$" is printed on the cell $C_k$ at time $t = 3(i+1)^2 - 4(i+1) + 4 = 3i^2 + 2i + 3$.
- **$S_2$:** At time $t = 3i^2 + 2i$, the both c'- and d-waves meet on the cell $C_k$, where $k = i^2+2i-2$. Let $\Delta_1$, $\Delta_2$ be any integer such that $0 \leq \Delta_1 \leq 4i+3$, $0 \leq \Delta_2 \leq 2i+1$, respectively. At time $t = 3i^2 - 2i + \Delta_1$, the 1/2-speed c-wave stays at $C_{i^2+\lfloor \Delta_1/2 \rfloor}$. On the other hand, the 1/1-speed d-wave stays at $C_{i^2+\Delta_2}$ at step $t = 3i^2 + 2 + \Delta_2$. Therefore, the distance between cells where the both c- and d-waves are staying at step $t = 3i^2 + 2i$ is 2. In order to make those two waves have contact at this step, we introduce a new wave. The c-wave being propagated generates at every two steps a left-traveling 1/1-speed tentacle-like wave that will disappear one step later after its emergence. The signal is referred to as the *c'-wave.* The c'-wave acts as a look-ahead signal that notifies the d-wave of its timing of f-wave generation. At time $t = 3i^2 + 2i$, the both c'- and d-waves meet on the cell $C_k$, where $k = i^2 + 2i - 2$. When the two waves meets, the f-wave is generated there simultaneously.
- **$S_3$:** The left-traveling 1/1-speed f-wave, generated at $C_{i^2+2i-2}$ at step $t = 3i^2+2i$, meets the 1/3-speed e-wave on $C_\ell$, $\ell = i^2 + i - 1$ at step $t = 3i^2 + 3i - 1$. This statement can be easily proved by a simple calculation. The g- and u-waves will be generated simultaneously on the cell in which the f- and e-waves meet.
- **$S_4$:** The u-wave remains at $C_\ell$, $\ell = i^2 + i - 1$ for only four steps, and then generates a v-wave at step $t = 3i^2 + 3i + 3$.
- **$S_5$:** The g-wave hits the right end of $S_i$ at step $t = 3(i+1)^2 - 2(i+1) = 3i^2 + 4i + 1$ and generates the c-wave for $S_{i+1}$.
- **$S_6$:** The v-wave hits the left end of $S_i$ and generates the first a-wave in $S_i$ at step $t = 3i^2+4i+2$. The first a-wave hits the right end of $S_i$ at step $t = 3(i+1)^2 = 3i^2 + 6i + 3$ and initiates the generation of d- and e-waves at step $t = 3(i+1)^2 + 2 = 3i^2 + 6i + 5$. The a-wave for the 2nd, 3rd, ... reciprocations in $S_i$ are generated at the same timing, as is shown in Fig. 4.

Thus, the partition setting for the right end of $S_i$ $(i \geq 3)$ is made inductively. The first two markings on cells $S_1$ and $S_2$ at times $t = 7$ and 18, respectively, and the generation of c-, d- and e-waves at the left end of $S_2$ at steps 8 and 14 are realized in terms of finite state descriptions. Thus, we can set up those entire partitions inductively in time.

In addition to Lemma 2, the generation of a- and b-waves and a number of additional signals in $S_1$ and $S_2$, as shown in Fig. 5, are also implemented in terms of finite state descriptions. Figure 5 is our final time-space diagram for the real-time prime generation algorithm. We have implemented the algorithm on a computer. Each cell has 34 internal states and 71 transition rules. The transition rule set is given in Table 1. We have tested the validity of the rule set from $t = 0$ to $t = 20000$ steps. In Fig. 6, we show a number of snapshots of the configuration from $t = 0$ to 50. The readers can see that the first fifteen primes can be generated in real-time by the left end cell. Based on Lemmas 1 and 2, we obtain the following theorem.

**[Theorem 3]** A prime sequence can be generated by a $CA_{1-\mathrm{bit}}$ in real-time.

# 4 Conclusions

A sequence generation problem on a special restricted class of cellular automata having 1-bit inter-cell communications ($CA_{1-\mathrm{bit}}$) has been studied. Several state-efficient real-time sequence generation algorithms for non-regular sequences have been proposed.

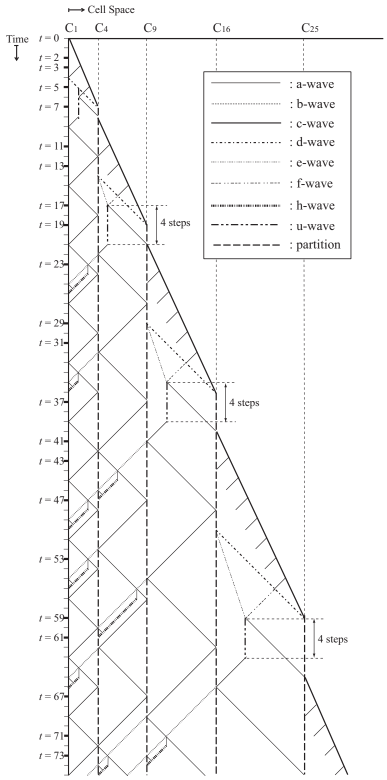


Figure 5: Time-space diagram for real-time prime generation.

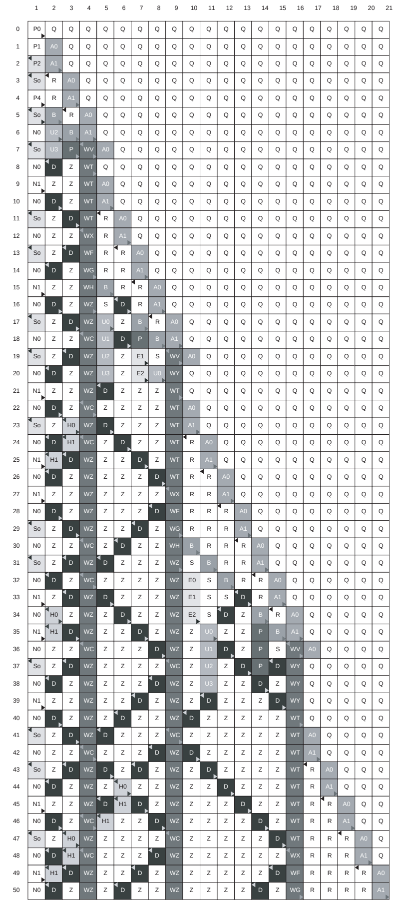

Figure 6: A configuration of real-time generation of prime sequences on the $CA_{1-\text{bit}}$ constructed with 34 states.

# 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] C. R. Dyer: One-way bounded cellular automata. *Information and Control*, Vol.44, pp.261-281, (1980).

[3] 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).

[4] I. Korec: Real-time generation of primes by a one-dimensional cellular automaton with 11 states. *Proc. of 22nd Intern. Symp. on MFCS '97, Lecture Notes in Computer Science*, 1295, pp.358-367, (1997).

[5] J. Mazoyer: A minimal time solution to the firing squad synchronization problem with only one bit of information exchanged. Technical report of Ecole Normale Superieure de Lyon, No.89-03, pp.51, April, (1989).

[6] J. Mazoyer: On optimal solutions to the firing squad synchronization problem. *Theoretical Computer Science*, 168, pp.367-404, (1996).

[7] J. Mazoyer and V. Terrier: Signals in one-dimensional cellular automata. *Theoretical Computer Science*, 217, pp.53-80, (1999).

[8] A. R. Smith: Real-time language recognition by one-dimensional cellular automata. *J. of Computer and System Sciences*, 6, pp.233-253, (1972).

[9] V. Terrier: On real-time one-way cellular array. *Theoretical Computer Science*, 141, pp.331-335, (1995).

[10] V. Terrier: Language not recognizable in real time by one-way cellular automata. *Theoretical Computer Science*, 156, pp.281-287, (1996).

[11] S. L. Torre, M. Napoli and D. Parente: Synchronization of a line of identical processors at a given time. *Fundamenta Informaticae* 34(1-2), pp.103-128, (1998).

[12] H. Umeo: Cellular algorithms with 1-bit inter-cell communications. *Proc. of MFCS'98 Satellite Workshop on Cellular Automata* (Eds. T. Worsch and R. Vollmar), Interner Bericht 19/98, University of Karlsruhe, pp.93-104, (1998).

[13] H. Umeo and M. Inada: A design of cellular algorithms for 1-bit inter-cell communications. *Proc. of the Fifth International Workshop on Parallel Image Analysis, IWPIA'97*, pp.51-62, (1997).

[14] H. Umeo: Linear-time recognition of connectivity of binary images on 1-bit inter-cell communication cellular automaton. *Parallel Computing*, 27, pp.587-599, (2001).

[15] H. Umeo and N. Kamikawa: A design of real-time non-regular sequence generation algorithms and their implementations on cellular automata with 1-bit inter-cell communications. *Fundamenta Informaticae*, 52, No.1-3, pp.255-273, (2002).

[16] H. Umeo and N. Kamikawa: An infinite prime sequence can be generated in real-time by a 1-bit inter-cell communication cellular automaton. *Proc. of The Sixth Internal Conference on Developments in Language Theory*, (Eds. M. Ito and M. Toyama), LNCS 2450, pp.339-348, (2002).

[17] T. Worsch: Linear time language recognition on cellular automata with restricted communication, *Proc. of LATIN 2000:Theoretical Informatics* (Eds. G. H. Gonnet, D. Panario and A. Viola), LNCS 1776, pp.417-426, (2000).