INFORMATIQUE THÉORIQUE ET APPLICATIONS

V. TERRIER

Real time recognition with cellular automata : a meaningful example

Informatique théorique et applications, tome 27, nº 2 (1993), p. 97-120

<http://www.numdam.org/item?id=ITA_1993__27_2_97_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 27, n° 2, 1993, p. 97 à 120)

REAL TIME RECOGNITION WITH CELLULAR AUTOMATA: A MEANINGFUL EXAMPLE (*)

by V. TERRIER ([1])

Communicated by C. CHOFFRUT

Abstract. – *A. Hemmerling, answering a conjecture of O. Ibarra, S. Kim and S. Moran, has shown using a sequential model, that the set of the strings over the alphabet* $\{0, 1\}$ *such that the number of* 1*'s is equal to the number of* 1*'s in the binary representation of their length is recognized in real time by a cellular automaton. Here, we present an alternative, self-contained cellular automata solution. Our construction points out the possibilities of self-organization in cellular automata and exploits the notion of signal to manage the information.*

Résumé. – *A. Hemmerling, répondant à une conjecture de O. Ibarra, S. Kim et S. Moran, a montré avec un modèle séquentiel, que l'ensemble des mots sur l'alphabet* $\{0, 1\}$ *dont le nombre de* 1 *est égal au nombre de* 1 *de la représentation en binaire de leur longueur est reconnaissable en temps réel par un automate cellulaire. Ici, nous présentons une autre solution qui a le mérite d'être intrinsèque aux automates cellulaires. Notre construction illustre les possibilités d'auto-organisation des automates cellulaires et notamment exploite la notion de signal pour gérer l'information.*

1. INTRODUCTION

The computing capabilities of one-dimensional cellular automata (CA for short) are known to be powerful since they can simulate any Turing machine in an at least as efficient way. This parallel model allows us to distribute and synchronize the information in a very efficient way. A lot of interest focuses on the lower class of complexity: the real time CA languages [1, 2, 3, 6, 7, 8]. Because of the combinatorial capabilities of CA, it is easier to talk about its power than its limitations. Nevertheless Smith [9] showed that a CA which works in time $f(n)$, can be simulated by a Turing machine

(*) Received June 1991, final version May 1992.
([1]) LAIAC-Université de Caen, 14032 Caen cedex.

0988-3754/93/02 97 24/$4.40/

in space $f(n)$ and time $f^2(n)$; so there exist languages that are not accepted in real time by any CA. More precisely, O. Ibarra, S. Kim, S. Moran [8] conjectured that languages accepted by CA in linear time but not real time exist. As a possible candidate, they proposed the set L of binary strings ω whose number of 1's is equal to the number of 1's in the binary representation of their length. However, A. Hemmerling [6] disproved the conjecture by showing that this language is recognized in real time. For that purpose, he used a special type of sequential machine, the full scanning Turing machine (STM) introduced by O. Ibarra, S. Kim, S. Moran [8] as a variation of the Turing machine which has a time complexity equivalent to the time complexity of CA's. In order to prove that the set L can be accepted by a STM in real time, A. Hemmerling described an encoding of the number of 1's in the binary representation of the length of the strings ω, which uses three functions defined by simultaneous recursions.

In this paper, we construct an alternative, self-contained CA solution. The basic features of our solution are different from those of Hemmerling since it relies on signal methods. Indeed, specific algorithms on CA are not generally described by their state transition function but by a more amenable object: the signals [4, 5]. The signals appear intuitively as an encoding of the information and its moves. In the following algorithm a network of signals is set up by one recursive call of a basic signal: the signal which marks the time-axis at times 2^n. This network is used in two manners:

– as an encoder: the number of 1's in the binary representation of the length of the strings ω is given by the intersections of this network and the "end of line" signal.

– as a dispatcher: the information is distributed according to the lines of the network.

2. PRELIMINARIES

Given $\omega = a_1 a_2 \ldots a_n$ with $a_i \in \{0, 1\}$ for $i \in \{1, \ldots, n\}$, we denote its *length* by $|\omega|$, the binary representation of the length of ω by $|\omega|_2$ and the number of 1's by $\#_1(\omega)$. More generally, the binary representation of an integer k is $\mathrm{bin}(k)$.

We quickly recall the behavior of the general CA's. A CA is a one dimensional-array of identical cells numbered $0, 1, \ldots$ from left to right and working synchronously. Each cell communicates with both left and right neighbors. The state of a cell c at time t depends on the states of the cells

$c-1$, c, $c+1$ at time $t-1$. The set of the states is finite and a subset of *accepting* states is distinguished. Here, we consider states as signal emissions. So, the signal emission of a cell c at time t depends on signals received from the cells $c-1$, c, $c+1$ at time $t-1$. The cell 1 communicates with the outside.

In particular, at time 0, the i-th bit of the input string ω is on the i-th cell, the remaining cells are in the quiescent state λ. At time 1, cells 1 and $|\omega|$ know upon receipt of the state λ, that they are the ends of the significant segment of the array.

We say that a CA recognizes a language L in real time, if it accepts the strings $\omega \in L$ in $|\omega|$ steps, *i.e.*, if the cell 1 enters an accepting state at time $|\omega|$.

The problem consists of building a CA recognizing the language $L=\{\omega;\ \#_1(\omega)=\#_1(|\omega|_2)\}$ in real time.

Example: if $\omega=101$ then $\#_1(\omega)=2$ and $\#_1(|\omega|_2)=\#_1(11)=2$ thus $\omega \in L$. However, if $\omega=1001$ then $\#_1(\omega)=2$ and $\#_1(|\omega|_2)=\#_1(100)=1$ thus $\omega \notin L$.

We represent the evolution of a CA by means of a time-space diagram (cf. *fig.* 1). By unrolling the time, we construct a two-dimensional array. A position c along with a time t defines the *site* (c, t). To achieve a real time computation, the working area of a CA must be bounded by the signal travelling at speed 1 which leaves the end of the word ω at time 1 and reaches the cell 1 at time $|\omega|$. Observe that it is the fastest signal that the cell 1 can receive from the end of the word and which returns an information about the length of the word.

First, we set up signals which will intersect the signal defined above. The number of those intersections will correspond to the number of 1's in the binary representation of the length of ω. These signals are constructed via recursive calls of the procedure defined by C. Choffrut and K. Culik [2], to recognize the language $L=\{a^{2^n}, n \in N\}$. Then, we move the information about the 1's of the word ω in order to compare them with the signal intersections associated with the 1's of $|\omega|_2$.

3. PROCESSING THE LENGTH OF THE INPUT WORD

The purpose of this section is to interpret the 1's in the binary representation of the length of the input word by means of intersection of signal.

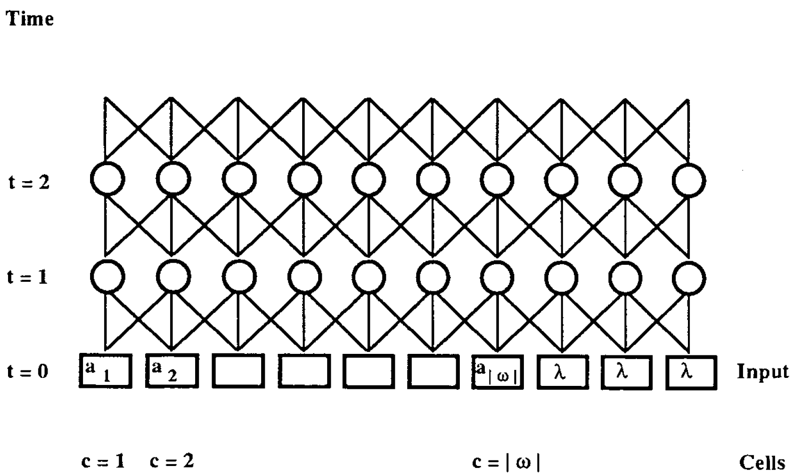


Figure 1

3.1. Preliminaries

C. Choffrut and K. Culik [2], showed that the language $L=\{a^{2^n};\ n\in N\}$ is recognized by a CA in real-time. Thus, in particular, this CA decides in real-time whether or not a string ω has exactly one 1 in the binary representation of its length, *i.e.* $\#_1(|\omega|_2)=1$.

They define two signals with different speeds (cf. *fig.* 2):

– a slow one which leaves the cell 1 at time 0, and runs from left to right at speed 1/3.

– a fast one which leaves the cell 1 at time 2 and runs at speed 1. It bounces between the slow signal and the cell 1. Thus, it zigzags between the cell 1 and the slow signal.

So, if the fast signal leaves the cell 1 at time 2^n, it reaches the slow signal on cell c at time t, where c and t are such that: $t=3.c=2^n+c$ thus $c=2^{n-1}$ and $t=3.2^{n-1}$. Then, the fast signal returns to cell 1 in c steps; so, it reaches cell 1 at time $t+c=2^{n+1}$. Thus, the fast signal draws in the time space diagram successive triangles of base 2^n, and marks cell 1 at times 2^n, for every n.

Let E_k be the ("end of line") signal which runs at speed 1 from right to left and reaches cell 1 at time $2k$. We assume that this signal leaves cell $k+1$ at time k.

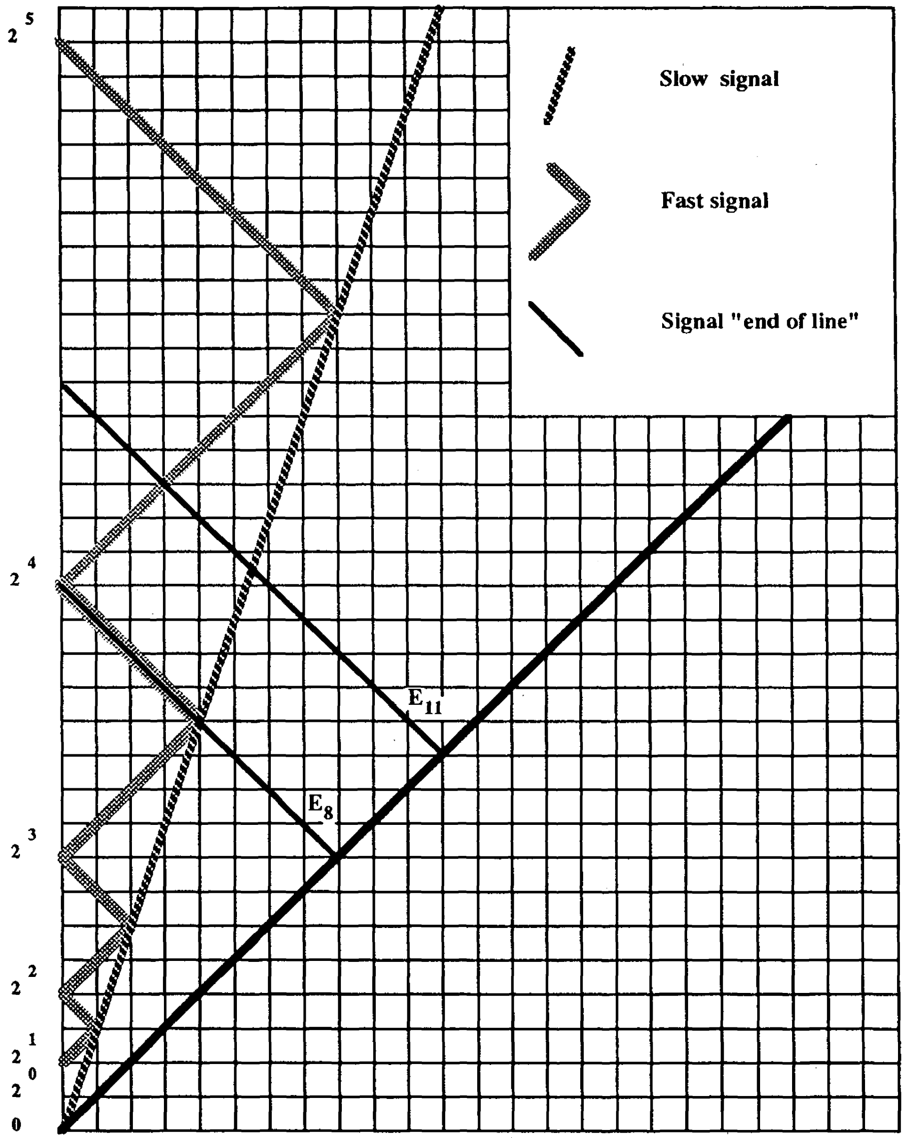


Figure 2

We observe that for all $k>1$:

– Either this signal E_k merges with the edge of a triangle: in this case, it corresponds to a string which has exactly one 1 in the binary representation of its length.

– Or it enters a triangle: in this case, it corresponds to a string which has at least two 1's in the binary representation of its length.

3.2. Recursive call of the procedure

In 3.1, we have seen that a signal E_k enters a triangle if k has at least two 1's in its binary representation. Let us be more precise (cf. *fig.* 3).

If a signal E_k goes inside a triangle whose base has length 2^n, then:

$$2k = 2^n + \rho \qquad \text{with} \quad 0 < \rho < 2^n,$$

and

$$\#_1 (\mathrm{bin}(2k)) = \#_1 (\mathrm{bin}(\rho)) + 1.$$

Thus, repeating one more time the procedure described in 3.1, inside the previous triangle, we can determine the integers which have exactly two 1's in their binary representation:

– The slow signal leaves cell 1 at time 2^n, runs at speed 1/3 and stops when it reaches the edge of the triangle.

– The fast signal leaves cell 1 at time 2^n+2, runs at speed 1, bounces between the slow signal and cell 1. Thus, it reaches cell 1 at times $2^n+2^2, \ldots, 2^n+2^i, \ldots, 2^n+2^{n-1}$, and finally merges with the edge of the triangle whose endpoint is cell 1 at time 2^{n+1}. In this way this new fast signal marks cell 1 at all times which have exactly two 1's in their binary representation. It also draws new triangles of base 2^i, where $0 < i < n$.

The slow signal which moves at speed 1/3 to the right is defined as follows: it remains on the same cell for two steps, and at every third step it moves to the right. At the slow signal initialization on cell 1, we can omit the first two steps. If cell 1 at time 2^n+2 does not receive a fast signal from the right, it sends a slow and a fast signal (this happens if cell 1 at time 2^n+2 is inside a triangle of base greater than 2). Thus, we will depict the useful part of the slow signal by a broken line.

Inside the first triangle, the signal E_k:

– either merges with the edge of a second triangle: it corresponds then to a string which has exactly two 1's in its binary representation.

– or it enters a second triangle: it corresponds then to a string which has at least three 1's in its binary representation.

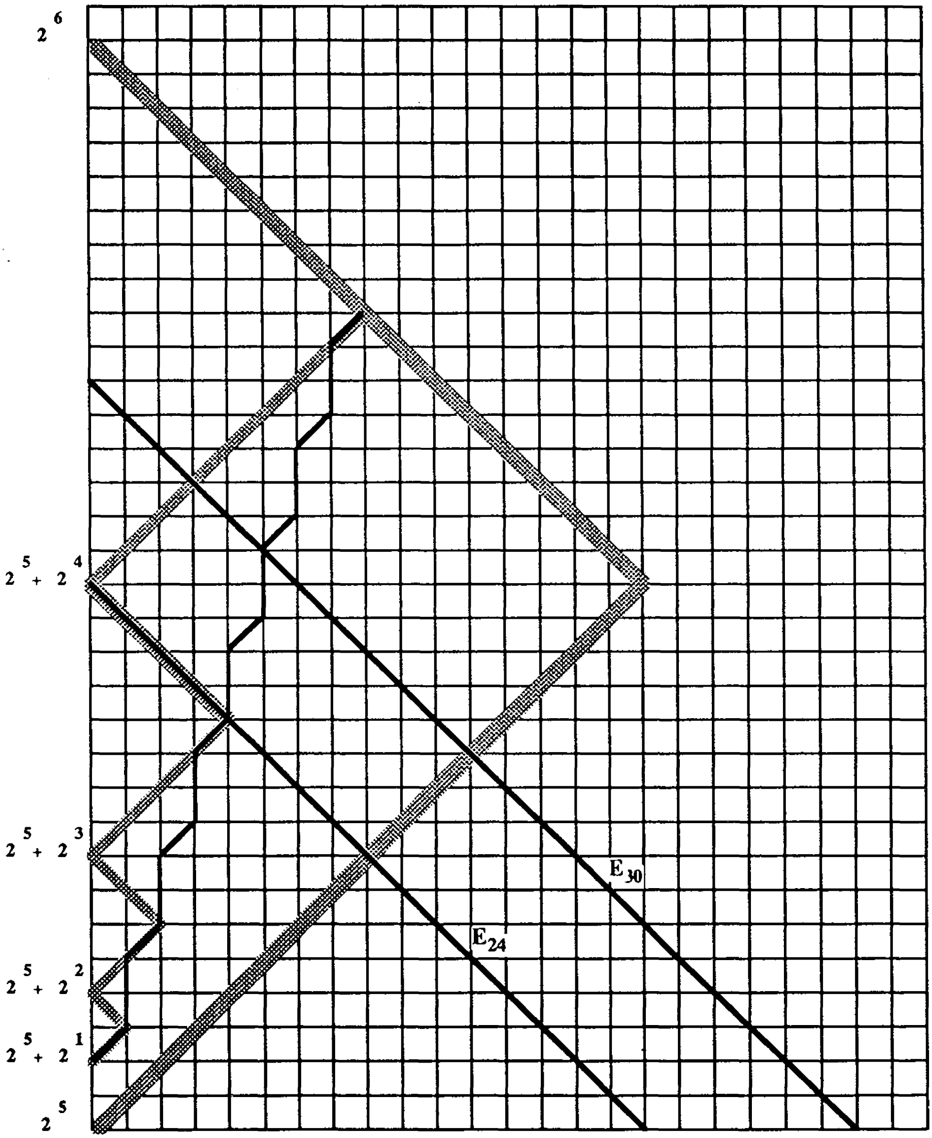


Figure 3

Next, by recursive calls of the procedure, inside each triangle of base 2^i, where $i>1$, we construct $i-1$ triangles of base 2^j, with $j=1, \ldots, i-1$. In this way we obtain *fig.* 4.

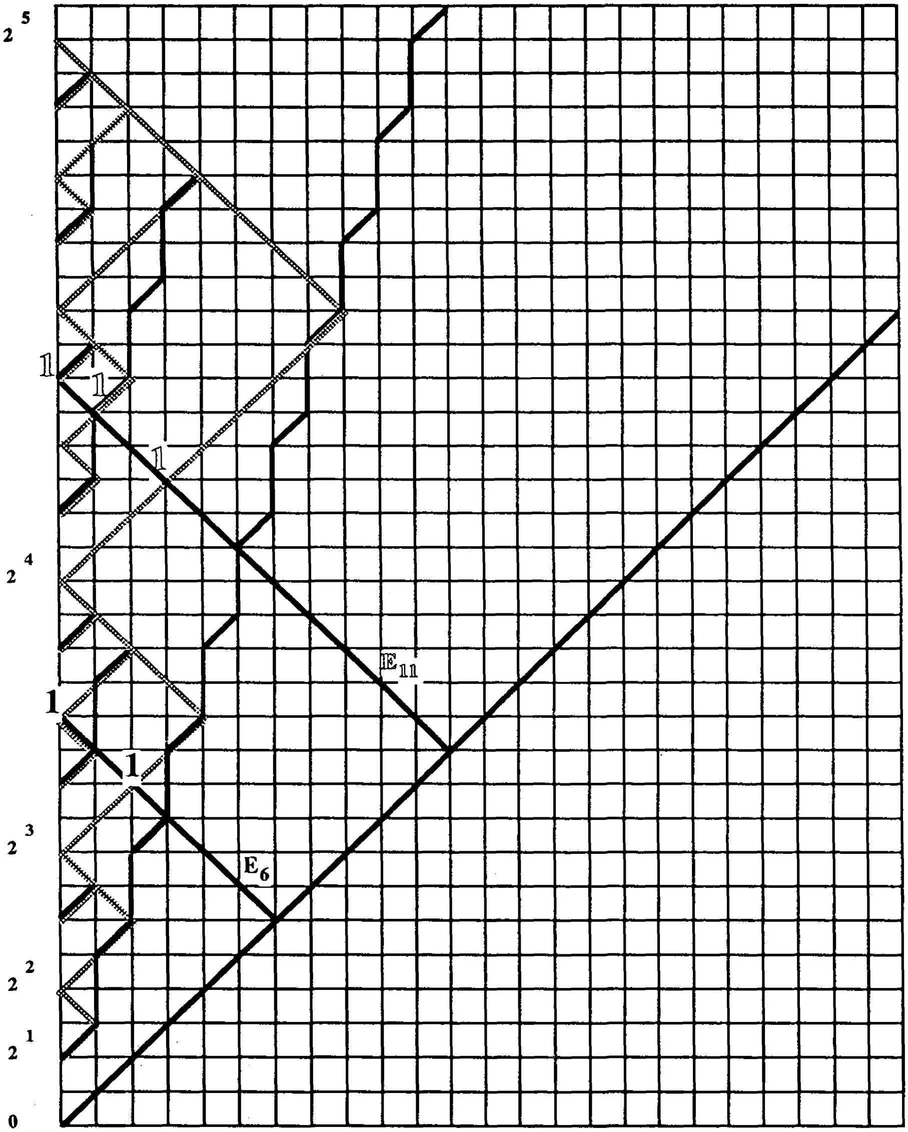


Figure 4

3.3. The sites associated with the 1's in the binary representation of an integer

Here we will prove that the number of 1's in the binary representation of an integer k corresponds to the number of times the signal E_k reaches a cell which sends a fast signal to its right.

Let F_{2i} be the set of sites (c, t) visited by the fast signal, between time $2i$ where it leaves cell 1, until the time where it reaches a slow signal. The sites of the fast signal are of the form $(1+u, 2i+u)$ with $u \geqq 0$. If $2i = 2^{b_h} + \ldots + 2^{b_2} + 2^{b_1}$, with $b_h > \ldots > b_1 > 0$, by construction the fast signal meets the slow signal which represents the line of slope 3 leaving the cell 1 at time $t = 2^{b_h} + \ldots + 2^{b_2}$. Thus, the fast signal and the slow signal meet on cell $1+u$ at time $t = 2^{b_h} + \ldots + 2^{b_2} + 2^{b_1} + u = 2^{b_h} + \ldots + 2^{b_2} + 3u$, so $u = 2^{b_1 - 1}$. Then $F_{2i} = \{(1+u, 2i+u), 0 \leqq u \leqq 2^{b_1 - 1}\}$. Thus the sites which send a fast signal to the right are the sites of F_{2i} excluding the last site. The set of these sites is denoted by S_{2i}.

The set of the sites visited by the "end of line" signal E_k which leaves cell $k+1$ at time k and runs at speed 1 from right to left is $E_k = \{(1+u, 2k-u), 0 \leqq u \leqq k\}$.

PROPOSITION: $\operatorname{Card}(\bigcup_i (S_{2i} \cap E_k)) = \#_1(\operatorname{bin}(k))$.

Let $2k = 2^{a_n} + \ldots + 2^{a_2} + 2^{a_1}$, so $\#_1(\operatorname{bin}(k)) = n$. Let $(c, t) \in S_{2i} \cap E_k$ with $2i = \Sigma\, 2^{b_r}$. Since $(c, t) \in S_{2i}$, then $(c, t) = (1+u, 2i+u)$, where $0 \leqq u < 2^{b_1 - 1}$. On the other hand, $(c, t) \in E_k$, then $(c, t) = (1+u, 2k-u)$, where $0 \leqq u \leqq k$. So $(c, t) \in \bigcup_i (S_{2i} \cap F_k)$ if and only if there exist i and u such that $2k = 2i + 2u$ and $0 \leqq u < 2^{b_1 - 1}$.

As $2u < 2^{b_1}$ and 2^{b_1} divides $2i$, we have $2u \equiv 2k \bmod (2^{b_1})$. Then if b_1 is such that $a_t < b_1 \leqq a_{t+1}$, we have $2u = \sum_{r=1}^{t} 2^{a_r}$ and $2i = \sum_{r=t+1}^{n} 2^{a_r}$ with $b_1 = a_{t+1}$. Thus, $\bigcup_i (S_{2i} \cap E_k) =$

$$\left\{(1, 2k), \left(1 + 1/2 \sum_{r=1}^{t} 2^{a_r}, 2k - 1/2 \sum_{r=1}^{t} 2^{a_r}\right) \text{ for } t = 1, \ldots, n-1\right\}$$

and $\operatorname{Card}(\bigcup_i (S_{2i} \cap E_k)) = n = \#_1(\operatorname{bin}(k))$

3.4. Description of the CA

Recall that we identify states with signal exchanges. Thus, in order to describe a cellular automaton which sets up the previous procedures, we have to specify emission and reception of signals. The different situations, required by our CA are depicted in *fig.* 5.

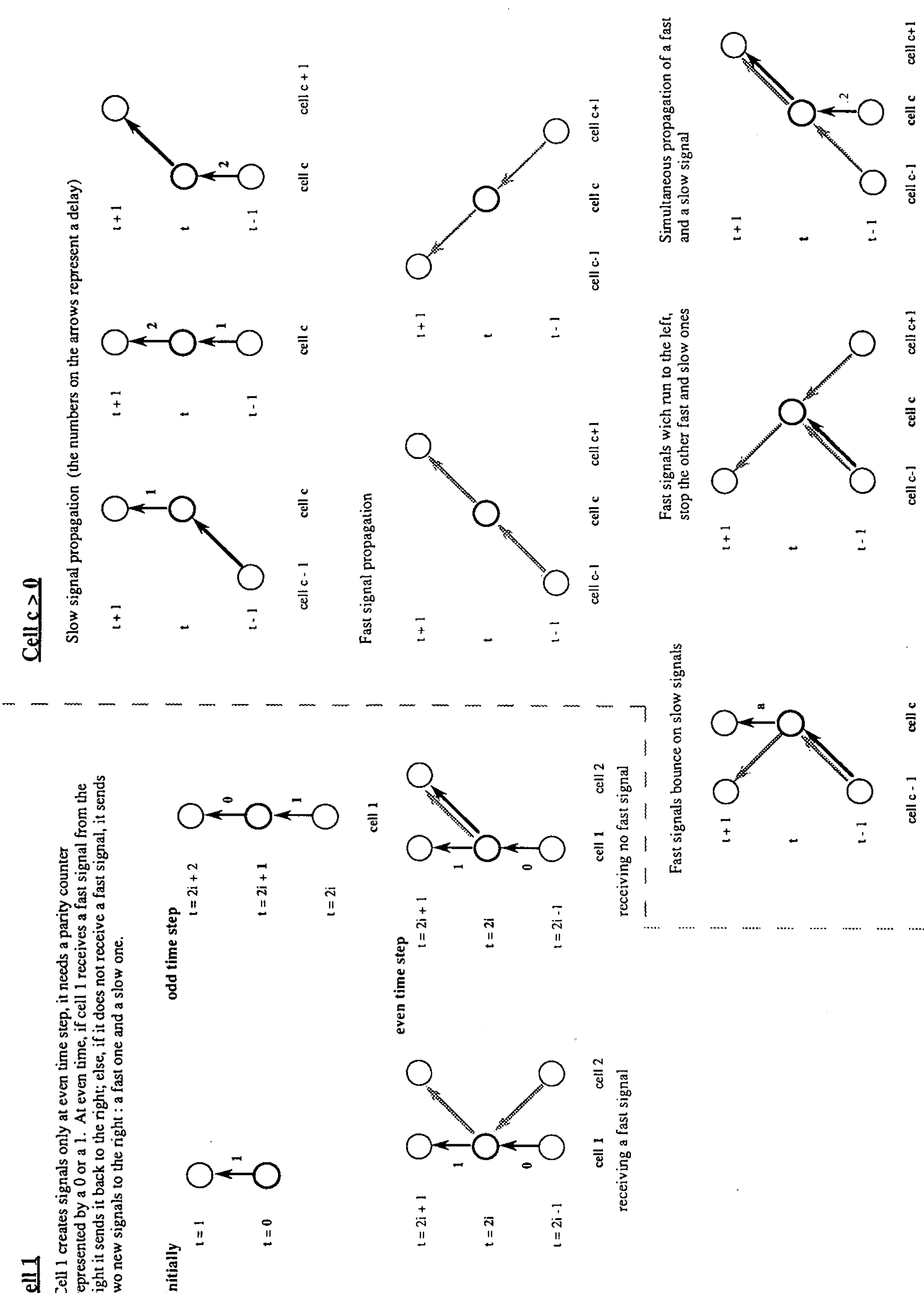

Cell 1
Cell 1 creates signals only at even time step, it needs a parity counter represented by a 0 or a 1. At even time, if cell 1 receives a fast signal from the right it sends it back to the right; else, if it does not receive a fast signal, it sends two new signals to the right : a fast one and a slow one.
Initially
t = 1
t = 0
odd time step
t = 2i + 2
t = 2i + 1
t = 2i
cell 1
even time step
t = 2i + 1
t = 2i
t = 2i - 1
cell 1
cell 2
receiving a fast signal
receiving no fast signal
Cell c > 0
Slow signal propagation (the numbers on the arrows represent a delay)
t + 1
t
t - 1
cell c - 1
cell c
cell c + 1
Fast signal propagation
cell c-1
cell c+1
Fast signals bounce on slow signals
Fast signals wich run to the left, stop the other fast and slow ones
Simultaneous propagation of a fast and a slow signal

Figure 5

4. PROCESSING THE BITS OF THE INPUT WORD

4.1. Moving the input word to the diagonal

The purpose of this subsection is to prepare the data. More precisely, we move the word to the diagonal of the space time diagram. This necessitates to solve a technical problem arising from the parity of the length.

Fig. 6 illustrates this section for different cases depending on the parity of $|\omega|$ and on the last bit of ω.

The 0-diagonal is the set of sites $(t+1, t)$.

In section 3, we showed how the fast signal E_k, emitted by cell $k+1$ at time k, can be used to determine the number of 1's in $\text{bin}(k)$. In fact, the signal E_k is the part of the fast signal which leaves cell $2k+1$ at time 0.

On the other hand, note that $\#_1(\text{bin}(2k)) = \#_1(\text{bin}(k))$

$$\text{and} \quad \#_1(\text{bin}(2k+1)) = \#_1(\text{bin}(k)) + 1$$

So the string of length $2k$ or $2k+1$ will be associated with the signal E_k. The end of the string is marked by the quiescent state λ. Thus, the first λ which is initially on the cell $2k+1$ if the length is $2k$, or on the cell $2k+2$ if the length is $2k+1$, must be received by the cell $2k+1$ at time 0 so that it can generate E_k.

Therefore, we take as initial time $t=-1$. We do not lose real time by choosing as initial time -1. Indeed, O. Ibarra, S. Kim, S. Moran [8] and C. Choffrut and K. Culik [2] have shown that the real time class is defined to within some additive constant.

At time -1, every cell sends its own input to its left neighbor and to itself.

We group the bits of ω as follows: we first add the consecutive bits of ω two by two. Then we send them to the left: so at time 0, $a_i + a_{i+1}$ is sent from cell i. By summing up the initial bits two by two, we have twice as much information as we really need. To screen that information we use the diagonal leaving cell 1 at time -1, which draws the line $(c+1, c-1)$. It stops on its cells the $a_{2i} + a_{2i+1}$'s, so that only the $a_{2i+1} + a_{2i+2}$'s remain and reach the first k cells of the 0-diagonal.

The cell $k+1$ of the 0-diagonal, from which the signal E_k starts, receives the information sent by the cell $2k+1$ at time 0. Thus:

– when the length is $2k$, the cell $2k+1$ at time 0 receives two λ and then sends a signal $\mathbb{E}$ at speed 1 which is as the signal E_k. This signal $\mathbb{E}$ expresses

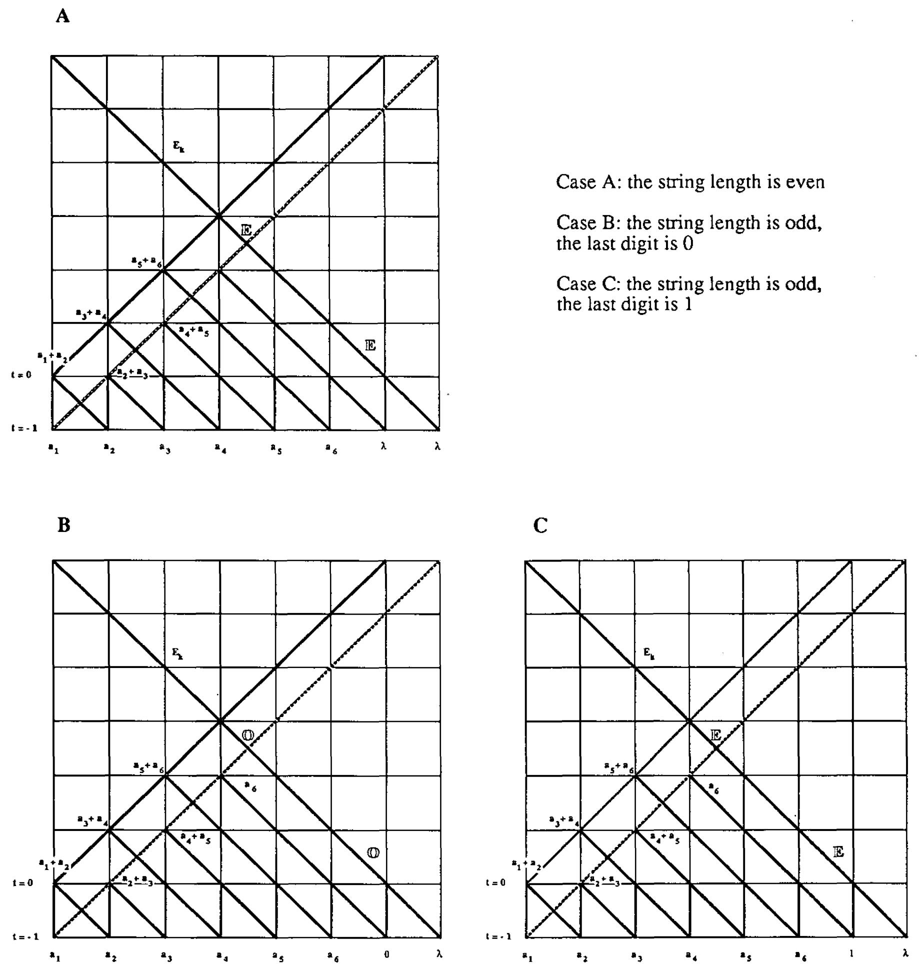


Figure 6

the fact that the number of 1's in the binary representation of the length is exactly the number of 1's in the binary representation of k.

– When the length is $2k+1$, the cell $2k+1$ at time 0 receives a_{2k+1} and λ.

We will need to distinguish two cases according to the value of the last bit a_{2k+1}. If $a_{2k+1}=1$, then as in the even case the cell $2k+1$ at time 0 sends a signal $\mathbb{E}$: indeed, if the string $\omega = a_1 a_2 \ldots a_{2k+1}$ is such that $a_{2k+1} = 1$, then

$$\#_1 (a_1 a_2 \ldots a_{2k+1}) = \#_1 (\mathrm{bin}(2k+1))$$

if and only if

$$\#_1 (a_1 a_2 \ldots a_{2k}) = \#_1 (\mathrm{bin}(2k))$$

and thus this case can be treated as the even case. If $a_{2k+1} = 0$, then the cell $2k+1$ at time 0 sends a signal $\mathbb{O}$ which expresses the fact that the number of 1's in the binary representation of the length is exactly 1 plus the number of 1's in the binary representation of k.

In this way, the signal E_k is represented by $\mathbb{E}$ or $\mathbb{O}$.

4.2. Comparison between the number of 1's in ω and the number of 1's in the binary representation of its length

The only signals emitted by the cells of the 0-diagonal are 0, 1, 2, $\mathbb{O}$, $\mathbb{E}$ and they are distributed as follows:

– at time k, the cell $k+1$ emits to the cell k a signal $\mathbb{O}$ if $|\omega|$ is odd and a_{2k+1} is 0. In other cases, the cell $k+1$ emits to the cell k a signal $\mathbb{E}$.

– for $i \in \{1, \ldots, k-1\}$, at time i, the cell $i+1$ emits to the cell i a signal 0, 1 or 2 according to the value 0, 1 or 2 of $a_{2i+1} + a_{2i+2}$.

– at time 0, the cell 1 emits to the cell 1 a signal 0, 1 or 2 according to the value 0, 1 or 2 of $a_1 + a_2$.

The special case where $|\omega|$ is 1 is easily checked.

In this way, the input ω is encoded on the 0-diagonal by the emission of signals. This information about the 1's of ω is sent to the left and sometimes is moved to the right in order to reach a site associated with an occurrence of 1 in $|\omega|_2$.

4.2.1. Moving the 1's of the input word ω

NOTATIONS (cf. *fig.* 7).

– We denote by T_{2i} the segment of the $2i$-diagonal between the cell 1 at time $2i$ and the "end of line" signal, excluding the site on the "end of line" signal, *i.e.*,

$$T_{2i} = \{(1+u,\ 2i+u),\ 0 \leqq u < |\omega|/2 - i\}$$

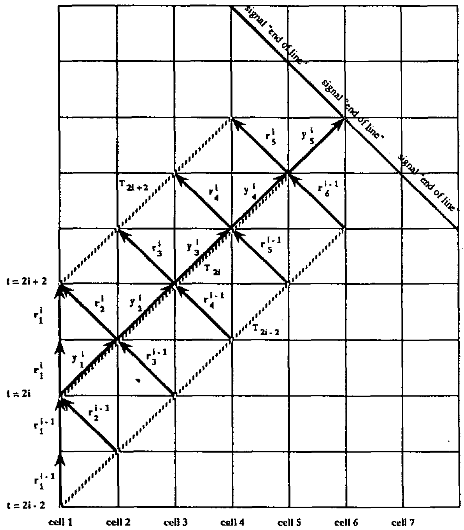


Figure 7

– We denote by r_m^i the signal 0, 1 or 2 sent by the cell m of T_{2i} at time $2i+m-1$ to the cell $m-1$ of T_{2i+2} if $m>1$, to the cell 1 of T_{2i+2} if $m=1$.

– We denote by y_m^i the signal 0, 1 or 2 sent by the cell m of T_{2i} to the cell $m+1$ of T_{2i}.

Initially, the $\lfloor |\omega|/2 \rfloor$ cells of T_0 send the signals $r_m^0 = a_{2m-1} + a_{2m}$ to the $\lfloor |\omega|/2 \rfloor - 1$ cells of T_2. Thus, $\sum_{m=1}^{[|\omega|/2]} r_m^0 = \#_1(\omega)$.

All cells of T_{2i} are associated with none of 1's in $|\omega|_2$, therefore we must make sure that they do not lose information. It is necessary that the sum of the bits received by each cell equals the sum of the bits that it sends. This is captured by:

$$r_m^i + y_m^i = \begin{cases} r_{m+1}^{i-1} + y_{m-1}^i & \text{if} \quad m>1 \\ r_1^{i-1} + y_2^{i-1} & \text{if} \quad m=1 \end{cases}$$

Then

$$\#_1(\omega)=\sum_{m=1}^{[|\omega|/2]} r_m^0 = r_1^1+y_1^1+\sum_{m=2}^{[|\omega|/2]-1}(r_m^1+y_m^1-y_{m-1}^1)$$

$$=\sum_{m=1}^{[|\omega|/2]-1} r_m^1 + y_{[|\omega|/2]-1}^1.$$

By recursion we have:

$$\#_1(\omega)=\sum_{m=1}^{[|\omega|/2]-i} r_m^i+\sum_{j=1}^{i} y_{[|\omega|/2]-j}^j = r_1^{[|\omega|/2]-1}+\sum_{j=1}^{[|\omega|/2]-1} y_{[|\omega|/2]-j}^j$$

The bits $r_1^{[|\omega|/2]-1}$ and $y_{[|\omega|/2]-j}^j$ are sent by the last cells of the T_{2i}'s and are received by the cells of the "end of line" signal (cf. *fig.* 8).

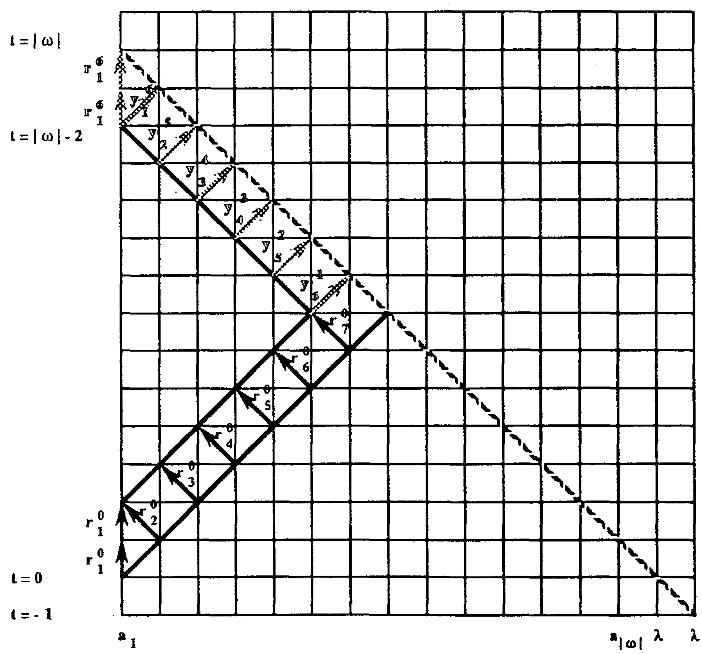


Figure 8

Then we would like that a cell of the "end of line" signal receives the bit 1 from a cell in T_{2i} if it is associated with an occurrence of 1 in $|\omega|_2$ and the bit 0 otherwise.

The information received by the cells of T_{2i} is sent according to the following principles. Note that the last cell of T_{2i} is required to send one 1 to the "end of line" signal if it is in S_{2i} (*i.e.* if it sends a fast signal to the right) and more precisely, if all cells of T_{2i} are cells of S_{2i}. So, we would like to set $y_m^i=1$ if the cell m sends a fast signal to the right, else $y_m^i=0$. However, if the sum $s=r_{m+1}^{i-1}+y_{m-1}^i$ is greater than 2 (or for $m=1$, $r_1^{i-1}+r_2^{i-1}$ is greater than 2), then the cell m of T_{2i} receiving s sends the excess to the right. Actually, the storage capacity of the r_m^i is bounded by 2. This can be summarized as follows:

For the cell 1 of T_{2i} (which always sends a fast signal to the right) we have:

$$\begin{array}{lll} \text{if} & r_1^{i-1}+r_2^{i-1}=0, & \text{then } r_1^i=y_1^i=0 \\ \text{if} & r_1^{i-1}+r_2^{i-1}=4, & \text{then } r_1^i=y_1^i=2 \\ \text{if} & 0<r_1^{i-1}+r_2^{i-1}<4, & \text{then } r_1^i=r_1^{i-1}+r_2^{i-1}-1 \quad \text{and} \quad y_1^i=1 \end{array}$$

For the cell m of T_{2i}

i) if it sends a fast signal to the right we have:

$$\begin{array}{lll} \text{if} & r_{m+1}^{i-1}+y_{m-1}^i=0, & \text{then } r_m^i=y_m^i=0 \\ \text{if} & r_{m+1}^{i-1}+y_{m-1}^i=4, & \text{then } r_m^i=y_m^i=2 \\ \text{if} & 0<r_{m+1}^{i-1}+y_{m-1}^i<4, & \text{then } r_m^i=r_{m+1}^{i-1}+y_{m-1}^i-1 \quad \text{and} \quad y_m^i=1 \end{array}$$

ii) if it does not send a fast signal to the right we have:

$$\begin{array}{lll} \text{if} & r_{m+1}^{i-1}+y_{m-1}^i\leqq 2, & \text{then } r_m^i=r_{m+1}^{i-1}+y_{m-1}^i \quad \text{and} \quad y_m^i=0 \\ \text{if} & r_{m+1}^{i-1}+y_{m-1}^i>2, & \text{then } r_m^i=2 \quad \text{and} \quad y_m^i=r_{m+1}^{i-1}+y_{m-1}^i-2 \end{array}$$

4.2.2. Proof of correctness

Here we give a formal proof of the correctness of the CA described in the previous subsection. We show that we can reduce the study to the case where the length of ω is even and that, in this case, the equality $\#_1(\omega)=\#_1(|\omega|_2)$ holds if and only if:

– the sites of the "end of line" signal which are associated with an occurrence of 1 in $|\omega|_2$, receive a single 1 from their left

– the other ones receive a single 0.

First, if the cells of the "end of line" signal which receive a single 1 are exactly the sites associated to an occurrence of 1 in $|\omega|_2$ and the other ones receive a single 0, then:

$$\#_1(|\omega|_2)=r_1^{[|\omega|/2]-1}+\sum_{j=1}^{[|\omega|/2]-1} y^j_{[|\omega|/2]-j},\ i.e.\ \#_1(|\omega|_2)=\#_1(\omega)$$

Conversely, if $\#_1(\omega)=\#_1(|\omega|_2)=n$, then there are exactly n indices $i>0$ such that $S_{2i}\cap E_{|\omega|/2}\neq\varnothing$, as we have shown in section 3.3. For these indices i, $S_{2i}\cap E_{|\omega|/2}$ are the sites of the "end of line" signal associated with an occurrence of 1 in $|\omega|_2$. Let $p_1, p_2, \ldots, p_n$ be these indices, with $0<p_1<p_2<\ldots<p_n$.
With the previous notations we have to show:

$$\begin{aligned} y^i_{|\omega|/2-i}&=1 \qquad \text{if} \quad i\in\{p_1, p_2, \ldots, p_n\}\\ &=0 \qquad \text{otherwise}\end{aligned}$$

We prove it by induction on i.

– We have $y^0_{|\omega|/2}=0$ and $0<p_1$

– Let i such that $p_t<i\leqq p_{t+1}$. (The limit cases $i\leqq p_1$ and $p_n<i$ are treated in the same way.)

By induction hypothesis, for all j where $j<i$, we have:

$$\begin{aligned} &y^j_{|\omega|/2-j}=1 \quad \text{if} \quad j\in\{p_1, p_2, \ldots, p_t\},\\ &y^j_{|\omega|/2-j}=0 \qquad \text{otherwise}\end{aligned}$$

Then, the sum of the bits arriving at the cells of T_{2i} is:

$$\sum_{m=1}^{|\omega|/2-i+1} r_m^{i-1}=\#_1(|\omega|_2)-\sum_{j=1}^{i-1} y^j_{|\omega|/2-j}=n-t$$

Now we show that $y^i_{|\omega|/2-i}=1$ if $i=p_{t+1}$ (case 1), and

$$y^i_{|\omega|/2-i}=0 \qquad \text{if} \quad i<p_{t+1}\ \text{(case 2).}$$

CASE 1: $i=p_{t+1}$

Then all cells of T_{2i} send a fast signal to the right, so:

– If $0<r_1^{i-1}+r_2^{i-1}<4$
  then $r_1^i=r_1^{i-1}+r_2^{i-1}-1$ and $y_1^i=1$
  and for $1<m\leqq|\omega|/2-i$, $r_m^i=r_{m+1}^{i-1}$ and $y_m^i=1$

– If $r_1^{i-1}+r_2^{i-1}=0$

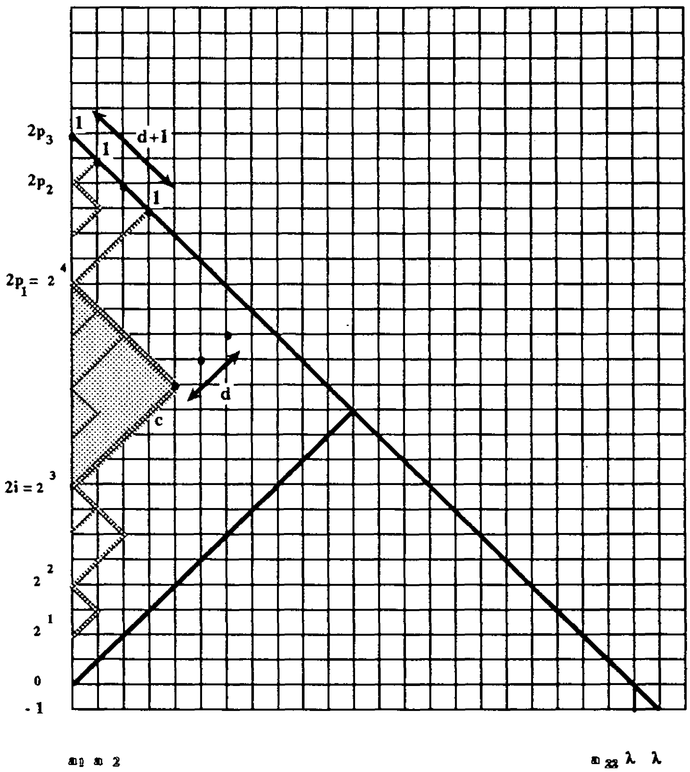


Figure 9

then let u be the smallest m such that $r_{m+1}^{i-1} > 0$.

Such an u exists since $\sum_{m=1}^{|\omega|/2-i+1} r_m^{i-1} = n - t \geqq t + 1 - t = 1$

Then for $m < u$, $r_m^i = 0$ and $y_m^i = 0$

for $m = u$, $r_u^i = r_{u+1}^{i-1} - 1$ and $y_u^i = 1$

for $u < m \leqq |\omega|/2 - i$, $r_m^i = r_{m+1}^{i-1} - 1$ and $y_m^i = 1$

– If $r_1^{i-1} + r_2^{i-1} = 4$

then let u be the smallest m such that $r_{m+1}^{i-1} < 2$. Such an u exists since

$$\sum_{m=1}^{|\omega|/2-i+1} r_m^{i-1} = n - t \leqq (|\omega|/2 - i + 1) < 2(|\omega|/2 - i + 1)$$

Then for $m<u, r_m^i=2$ and $y_m^i=2$

for $m=u, r_u^i=r_{u+1}^{i-1}+1$ and $y_u^i=1$

for $u<m\leqq|\omega|/2-i, r_m^i=r_{m+1}^{i-1}-1$ and $y_m^i=1$

Thus, in each case we have $y_{|\omega|/2-i}^i=1$

CASE 2: $i<p_{t+1}$ (cf *fig.* 9)

Let c be the number of the last cell of T_{2i} which sends a fast signal to the right. So the fast signal which leaves the cell 1 at time $2i$ changes direction on cell $c+1$ at time $2i+c$ and reaches cell 1 at time $2(i+c)$. And the part of this fast signal between $(c+1, 2i+c)$ and $(1, 2(i+c))$ stops all fast signals which run to the right, but it does not stop the fast signal travelling on the cels of $S_{2p_{t+1}}$ which reaches the "end of line" signal, by construction.

Therefore, $p_{t+1}\geqq c+i$.

The part of the "end of line" signal between the cell 1 and the $(t+1)$-th cell associated with an occurrence of 1 in $|\omega|_2$ (the cell which receives the fast signal travelling on the cells of $S_{2p_{t+1}}$) has exactly $n-t$ cells associated to an occurrence of 1 in $|\omega|_2$. As the number of cells in this part of the "end of line" signal is equal to $|\omega|/2-p_{t+1}+1$, we have $|\omega|/2-p_{t+1}+1\geqq n-t$.

Let $d=|\omega|/2-i-c$ be the number of cells of T_{2i} which do not send a fast signal to the right (and thus which are not required to send one 1 to the right), then it follows:

$$d+1\geqq|\omega|/2-p_{t+1}+1\geqq n-t$$

The sum of the bits received by these d cells of T_{2i} is:

$$y_c^i+\sum_{m=2+c}^{|\omega|/2-i+1} r_m^{i-1}\leqq n-t$$

Let u be the smallest m such that: $c<m\leqq|\omega|/2-i$ and $r_{m+1}^{i-1}+y_{m-1}^i\leqq 2$. Such an u exists since otherwise: $r_{m+1}^{i-1}+y_{m-1}^i>2$, $r_{m+1}^{i-1}\geqq 1$ and then $y_c^i+\sum_{m=2+c}^{|\omega|/2-i+1} r_m^{i-1}>2+d-1\geqq n-t$.

Finally for $c<m<u, r_m^i=2$ and $y_m^i=r_{m+1}^{i-1}+y_{m-1}^i-2$

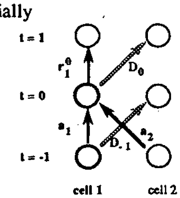
for $m=u, r_u^i=r_{u+1}^{i-1}+y_{u-1}^i$ and $y_u^i=0$

for $u<m\leqq|\omega|/2-i, r_m^i=r_{m+1}^{i-1}$ and $y_m^i=0$

Thus, in each case we have $y_{|\omega|/2-i}^i=0$. ■

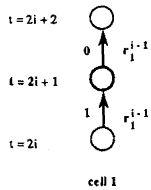
Cell 1

Initially

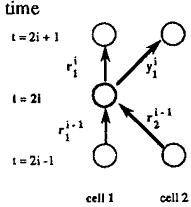


case 1: cell 1 is not on the "end of line" signal

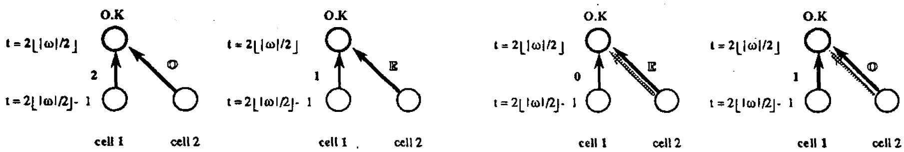
odd time



even time

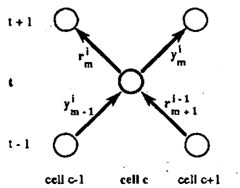


case 2: cell 1 is on the signal "end of line"



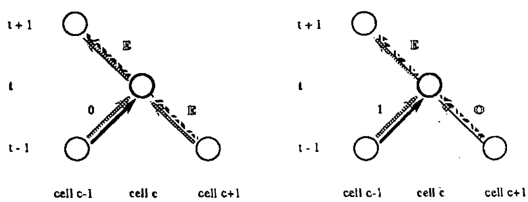
Cell c > 0

case 1: cell c is not on the signal "end of line"



cf page 9 for more details

case 2a: cell c is on the signal "end of line"
and is not associated with one 1 of the binary representation of the length



case 2b: cell c is on the signal "end of line"
and is associated with one 1 of the binary representation of the length

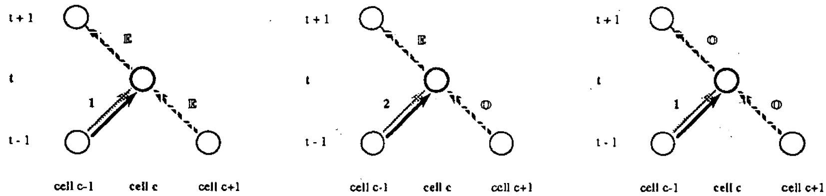


Figure 10

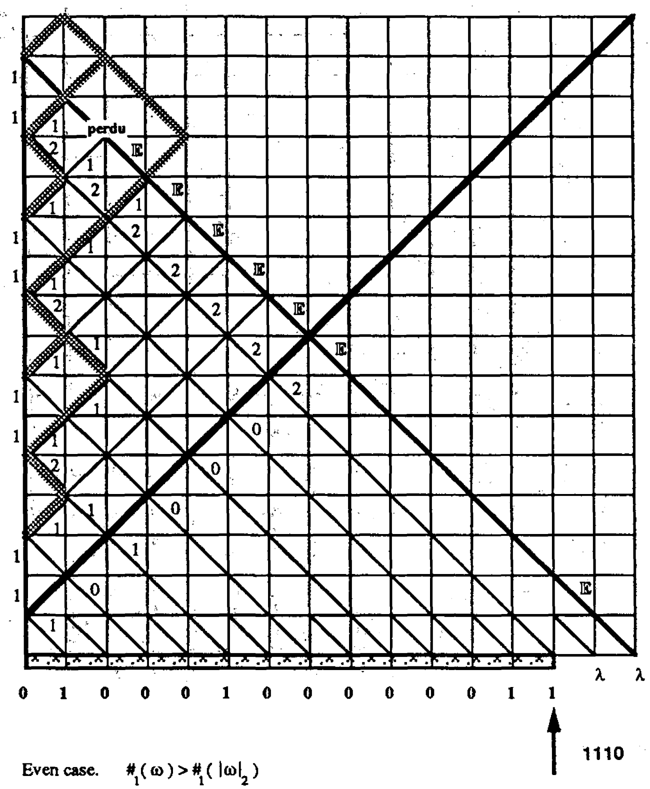


Figure 11

When the length $|\omega| = 2k+1$ is odd, we have seen in part 4.1 that the case where the last bit $a_{2k+1} = 1$ is treated as the even case. In the case where the last bit $a_{2k+1} = 0$, the "end of line" signal is labelled by $\mathbb{O}$. And $\#_1(|\omega|_2) = \#_1(\mathrm{bin}(k)) + 1$ corresponds to the $\#_1(\mathrm{bin}(k))$ cells of the "end of line" signal associated with an occurrence of 1 in $\mathrm{bin}(k)$, and by the label $\mathbb{O}$ of the signal "end of line" for the extra 1. So the comparison process is as in the even case, except that the label $\mathbb{O}$ erases one 1 and becomes $\mathbb{E}$.

We collect the result of the comparison process on the "end of line" signal: at time $2k$, cell 1 enters an accepting state if each cell of the "end of line" signal has received a bit 0 or 1 whether or not it is associated with an occurrence of 1 in $\mathrm{bin}(k)$.

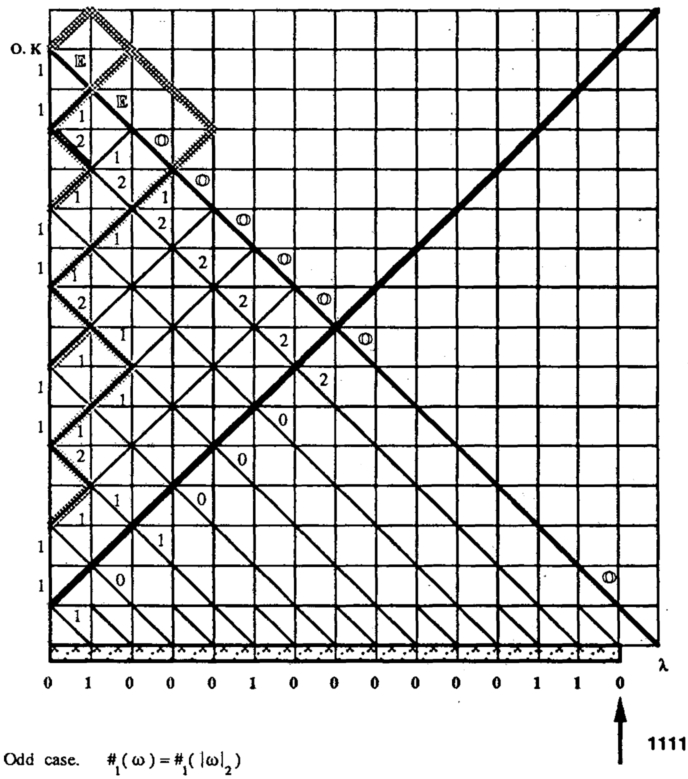


Figure 12

Description of the CA: the different situations required by the CA are listed on *fig.* 10.

Different examples are given in the *fig.* 11, 12, 13.

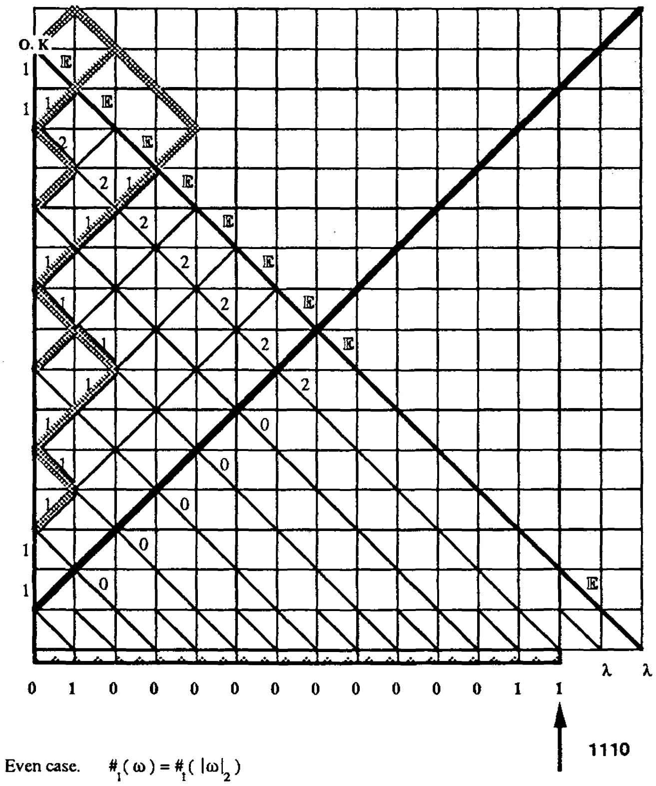


Figure 13

ACKNOWLEDGEMENTS

I thank Professor C. Choffrut for his indispensable help in revising the manuscript of this paper.

REFERENCES

1. W. Bucher and K. Culik, On real time and linear time cellular automata, *RAIRO Inform. Théor.*, *18*, (4), 1984, pp. 307-325.
2. C. Choffrut, K. Culik II, On real-time cellular automata and trellis automata, *Acta Inform.*, *21*, 1984, pp. 393-407.
3. S. N. Cole, Real-time computation by n-dimensional iterative arrays of finite-state machine, *IEEE Trans. Comput.*, *C-18*, 1969, pp. 349-365.
4. K. Culik, Variations of the firing squad problem and applications, *Information Processing Letters*, *30*, 1989, pp. 153-157, North-Holland.

5. P. C. Fischer, Generation of primes by a one-dimensional real-time iterative array, *J. ACM*, *12*, 1965, pp. 388-394.
6. A. Hemmerling, Real-time recognition of some language by trellis and cellular automata and full scan Turing machines, *EATCS*, n° 29, June 1986, pp. 35-39.
7. O. Ibarra and T. Jiang I., Relating the power of cellular arrays to their closure properties, *TCS 57*, 1988, pp. 225-238.
8. O. Ibarra, S. M. Kim, S. Moran, Sequential machine characterizations of trellis automata and applications, *SIAM J Comput.*, Vol. 14, 1985, n° 2, pp. 426-447.
9. A. R. Smith, Cellular automata theory, *Technical Report* 2, Standford University, 1969.