Synchronization of a line of finite automata with nonuniform delays

Jacques Mazoyer

Laboratoire de l'Informatique du Parallélisme

Ecole Normale Supérieure de Lyon
Unité de recherche associée au CNRS n°1398

Synchronization of a line of finite automata with nonuniform delays

Jacques Mazoyer

December 1994

Research Report N$^{\rm O}$ 94-49

Ecole Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : (+33) 72.72.80.00   Télécopieur : (+33) 72.72.80.80
Adresse électronique : lip@lip.ens−lyon.fr

Synchronization of a line of finite automata with nonuniform delays

Jacques Mazoyer

December 1994

Abstract

We study the Firing Squad Synchronization Problem with non uniform delays in the case of a line of cells. The problem was solved in the general case by T. Jiang in time Δ^3. In the case of the line, we improve his result, obtaining the Δ^2. We observe that there does not exist an optimal solution. We also note that the strategy, used here, is the general strategy (Waksman's one) and thus, even in this case, we can break the line in its middle.

Keywords: Automata, synchronization, optimality

Résumé

Le problème de la synchronisation d'un graphe avec des délais de transmission non uniformes a tété récemment résolu par T. Jiang en temps Δ^3. Nous l'étudions dans le cas de la synchronisation d'une ligne d'automates finis. Nous obtenons dans ce cas une solution en temps Δ^2. Cette solution utilise la startégie de Waksman en coupant la ligne en ses milieux

Mots-clés: Automates, synchronisation, optimalité

Synchronization of a line of finite automata with nonuniform delays*

Jacques Mazoyer†‡

January 13, 1995

1 Introduction

The *Firing Squad Synchronization Problem* (in short FSSP) is a well known problem arising in the field of cellular automata, first introduced in literature by M. Minsky [7].

Let us recall the FSSP: is to construct a one dimensional cellular automaton with a neighborhood of three cells such that: whatever the number n of cells is, the cellular automaton evolves from the initial configuration in which all cells are in a quiescent state (L) except the end left one (the general in state G) in such a way that all cells enter a special state (the Fire F) simultaneously and for the very first time.

Many authors have generalized this problem. In this paper, we consider three generalizations:

Synchronization of graphs

> Let Γ be a connected, non oriented graph of bounded degree d with a distinguished vertex G. To any node is attached a finite automaton. The automaton, attached to node V, communicates with any automaton attached to a node connected with V. FSSP is to construct a cellular automaton such that: whatever the number of vertices and of edges are, the cellular automaton evolves from the initial configuration in which all cells are in a quiescent state (L) except the automaton on G (the general) which is in state G, in such a way that all cells enter a special state (the fire F) simultaneously and for the very first time.

*This work was supported by the Programme de Recherches Coordonnées Mathématiques et Informatique and the Esprit Basic Research Action : "Algebraic and Syntactical Methods In Computer Sciences".

†Ecole Normale Supérieure de Lyon, Laboratoire de l'Informatique du Parallélisme, 46 Allée d' Italie, 69364 Lyon Cedex 07, FRANCE. e-mail: mazoyer@lip.ens-lyon.fr

‡Institut Universitaire de Formation des Maitres de Lyon, 5 rue Anselme, 69317 Lyon Cedex 04, FRANCE.

Synchronization of nonuniform lines

Let Λ be a one dimensional cellular automaton in which all cells make their state transition synchronously (the delay between two state transitions is said a unit of time), in which the delay between two adjacent cells is a fixed number of units of time, depends on the numbering of the left cell but keeps constant among the evolution of the cellular automaton. FSSP is to construct a cellular automaton such that: whatever the numbers of cells and of delays between two adjacent cells are, the cellular automaton evolves from the initial configuration in which all cells are in a quiescent state (say L) except the left end one (the general) which is in state G, in such a way that all cells enter a special state (the Fire F) simultaneously and for the very first time.

Synchronization of nonuniform graphs

The last generalization is obtained by mixing the two previous ones. It is the synchronization of a graph in which delays between cells depend on the edges between cells (this can be viewed as a synchronization of a connected, non oriented graph of bounded degree d in which edges are labeled by integers, indicating the delay of communication).

The first solutions to the FSSP (in time $3n$) are due to M. Minsky and J. Mac Carthy [7]. A minimal time solution needs $2n-2$ units of time: first minimal time solutions are due to A.Waksman [10] and R. Balzer [1]. A minimal state solution has at least 4 states [1] and there exists a solution with 6 states [5].

The first solution to the problem of synchronization of graphs is due to P. Rosensthiel [8]. Synchronization of families of graphs was especially studied by K. Kobayashi [4].

The problem of synchronization of nonuniform lines is due to V. Varkhavsky, V.B. Marakhovsky and V.A. Peschansky [9], who obtained synchronization of such lines when all delays are equal: in this case, the time of synchronization is $O(2n\delta)$ where δ is the delay between two cells.

More recently, T. Jiang [2] [3] has obtained a solution to the problem of synchronization of nonuniform graphs of bounded degree. The synchronization time is in $O(\Delta^3 + \tau_{max})$ where τ_{max} is the maximum delay of any single link and Δ is the delayradius (let us call $d(G,V)$ the minimal number of units of time necessary for the general to send a message to vertex V, then Δ is the maximum of $d(G,V)$ for any vertex V of Γ different of G).

Rosenstiehl's method is to construct in a graph of bounded degree a path which contains all vertices and then to synchronize this path. Jiang's method is direct and in his paper he indicates that the usual strategy is not directly useful because a line cannot be easily broken in two equal parts.

In this paper, we present a solution to the problem of synchronization of nonuniform lines in time $O(\Delta^2)$ where Δ is the delayradius, that is the time for messages sent by the general to reach the right end cell. This solution uses Minsky's strategy and Jiang's features: the line is broken in two parts as close as possible of half the delayradius. By Minsky's strategy, we obtain two disjointed sublines: the left and right ones. After some delay (the left and right ones) the left part and the right one are synchronized independently in such a way that both synchronizations occur at the same time.

Using Rosenstiehl's method, this solution gives a solution to the problem of synchronization of nonuniform graphs in a time different from Jiang's one. The time of synchronization does not depend on the delayradius Δ or τ_{max}, it is in $O(\Phi^2)$ units of time where Φ is the sum of all delays between cells. Thus, time comparisons with Jiang's solution are not easy.

We shall use the two new following main features of Jiang's solution:

- a cell k can set up a loop between itself and another cell (say cell h): any information sent by cell k is reflected by cell h and when its reflection is reflected by cell k, this information is changed and sent to cell h, and so on ... In such a loop, cell k can make some computations on the delay between cells k and h.

- in usual solutions, all cells enter Fire as soon as they know that there is no reason not to do it. Now a cell computes how it does work in function of the expected synchronization time.

2 Definitions

Definition 1 *A finite communicating automaton $\mathcal{A}$ is a triplet (Q, S, δ) where:*

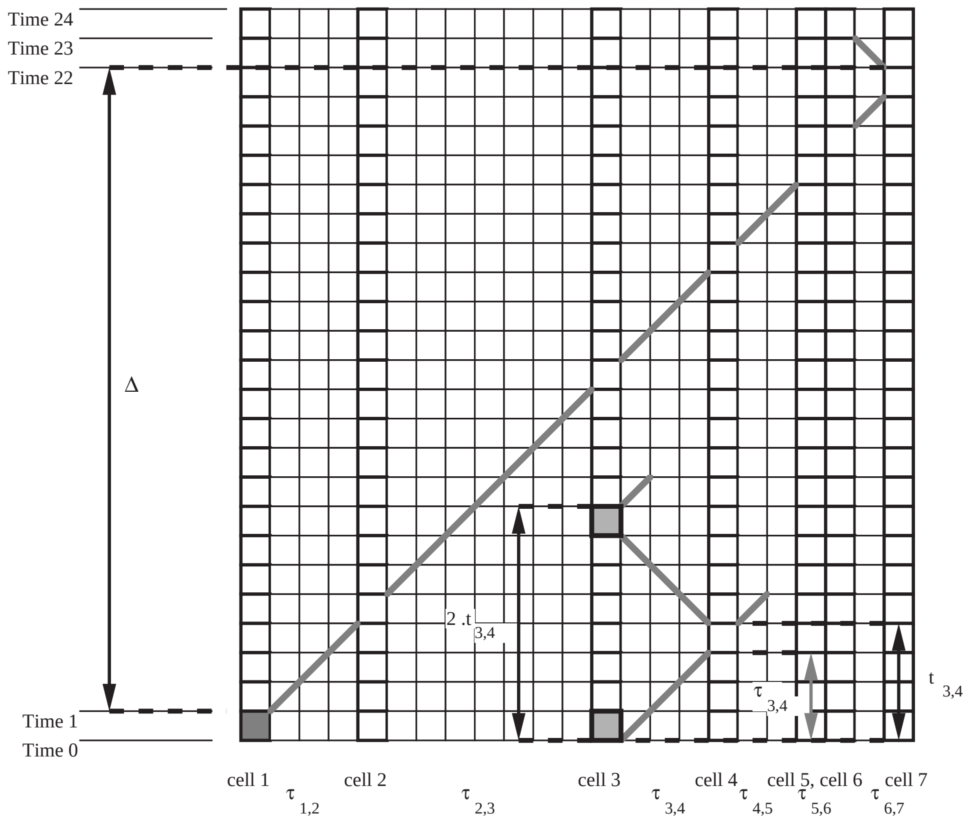
- *Q is the set of states, which contains three special states G (the General), L (the quiescent state) and F (the Fire);*

- *S is the set of signals, which contains two special signals S_T (the signal Time) and s_L (the quiescent signal);*

- *$\delta \ : \ S \times Q \times S \longrightarrow Q \times S$ is the state transition function.*

Definition 2 *1. A nonuniform line Λ is a couple $(n, \langle \tau_{1,2}, \ldots, \tau_{n-1,n} \rangle)$ where n is an integer ($n \geq 2$) and $\langle \tau_{1,2}, \ldots, \tau_{n-1,n} \rangle$ is a series of $n-1$ integers (which may be 0).*

2. The communicating delay between cell i and cell $i+1$ is $\tau_{i,i+1}$.



Case of a nonuniform line of 7 cells with delays :

$\tau_{1,2}$	= 3	$t_{1,2}$	= 4
$\tau_{2,3}$	= 7	$t_{2,3}$	= 8
$\tau_{3,4}$	= 3	$t_{3,4}$	= 4
$\tau_{4,5}$	= 2	$t_{4,5}$	= 3
$\tau_{5,6}$	= 0	$t_{5,6}$	= 1
$\tau_{6,7}$	= 1	$t_{6,7}$	= 2

$\Delta = 22$

Figure 1: Notations used in section 2

3. *The delay between cell i and cell $i+1$ is $t_{i,i+1} = \tau_{i,i+1} + 1$.*

4. *The delayradius of a nonuniform line Λ is* $\Delta_\Lambda = \sum_{1 \leq j \leq n} t_{i,i+1}$.

5. *A non uniform line of automata is a couple (A, Λ) where A is a finite communicating automaton and Λ a nonuniform line. Let us denote state (resp. signal) of the i^{th} automaton at time t by $\langle A, t \rangle_i$ (resp. $\ll A, t \gg_i$), we have: $(\langle A, t \rangle_i, \ll A, t \gg_i) =$*

$$\delta\left(\ll A, t - t_{i-1,i} \gg_{i-1}, \langle A, t-1 \rangle_i, \ll A, t - t_{i,i+1} \gg_{i+1}\right)$$

Definition 3 *A finite communicating automaton $\mathcal{A}$ is a solution to the problem of the synchronization of nonuniform lines in time $f(\Delta_\Lambda)$ if for any nonuniform line Λ:*

- *at time 0, cell 1 is in state G, emits signal S_T and all others cells (from 2 to n) are in state L emitting signal s_L,*
- *at time $f(\Delta_\Lambda)$, all cells enter state F simultaneously and for the very first time.*

The previous definition of a finite automaton is the same as usual but we distinguish letters of states from letters of sent messages (signals). We observe that although it receives two signals, a finite communicating automaton sends only one signal. In descriptions of finite communicating automata presented below, we shall often define S as a product $S_1 \times \ldots \times S_p$.

In a nonuniform line, we intuitively understand i as the numbering of a cell, and $t_{i,i+1}$ as the delay between cells numbered i and $i+1$ (the delay between cells $i+1$ and i is the same by hypothesis). In a nonuniform line of automata, if the finite automaton $\mathcal{A}$ which is on cell j, in state q, receives signal s_l from its left neighbor (machine $j-1$) and signal s_r from its right neighbor (machine $j+1$), it enters state q' and sends signal s' to its both neighbors (with $(q', s') = \delta(s_l, q, s_r)$).

The delay $t_{i,i+1}$ is the number of units of time needed by a signal sent by cell i to reach cell $i+1$. The delayradius is the number of units of time in which a signal sent by cell 1 is reflected by cell n. We observe that a signal sent by cell i, reflected by cell $i+1$, is reflected by cell i in $2t_{i,i+1}$ units of time. The figure 1 illustrates these remarks.

In the usual case of the Firing Squad, we have $\tau_{i,i+1} = 0$, but $t_{i,i+1} = 1$ ($\forall i \in \{1, \ldots, n-1\}$).

3 Breaking a nonuniform line in about its "middle"

Let Λ be a nonuniform line of n cells. For any k we define $\delta_{k,r}$ and $\delta_{k,l}$ (as right and left) by:

$$\delta_{k,r} = \sum_{i=k+1}^{i=n-1} t_{i,i+1}$$

$$\delta_{k,l} = \sum_{i=1}^{i=k-1} t_{i,i+1}$$

Obviously, the equality $\delta_{k,l} + t_{k,k+1} + \delta_{k,r} = \Delta$ holds.

Proposition 1 *There exists an automaton $\mathcal{K}$ such that on a nonuniform line Λ of n such automata $\mathcal{K}$ a special state BK (as "break") appears only on a cell, say cell k, at time $\Delta + \delta_{k,r} + t_{k,k+1}$ in the evolution of the initial line. In addition, we have:*

$$\delta_{k,l} \leq \lfloor \frac{\Delta}{2} \rfloor \leq \lceil \frac{\Delta}{2} \rceil \leq \delta_{k,r} + t_{k,k+1} \quad (1)$$

$$|\delta_{k,r} - \delta_{k,l}| \leq t_{k,k+1} \quad (2)$$

$$\delta_{k,r} \leq t_{k,k+1} + \delta_{k,l} \quad (3)$$

Proof

The figure 2 illustrates this proof.

First, a signal S_{Init} is sent by cell 1 through the line at time 0 and at maximal speed. It is reflected by cell n (the end right cell) at time $\Delta + \sum_{i=n}^{i=k+1} t_{i-1,i} - 1$ which is time $\Delta + \delta_{k,r} + t_{k,k+1} - 1$.

Second, another signal S_{Slow} is sent by cell 1 at time 1. But this signal goes through the line at speed $\frac{1}{3}$. In fact between cells h and $h+1$, this signal S_{Slow} starts from cell h, is reflected by cell $h+1$ as signal S_{Slow1} and is once more reflected by cell h as signal S_{Slow2}. Signals S_{Slow1} and S_{Slow2} are ignored by cells $h-1$ and $h+2$ (and, thus, suppressed). Finally, this signal S_{Slow2} is another time reflected by cell $h+1$ as signal S_{Slow} and, when cell h receives this third reflection, it suppresses it. This exchange of signals necessitates six states. Thus, starting from cell h at time t, signal S_{Slow} starts again from cell $h+1$ at time $t + 3t_{h,h+1}$. By this way, the signal S_{Slow} reaches cell i at time $3\delta_{i,l} - 1$.

A cell k enters the special state BK if it receives S_{Slow} and the reflection of S_{Init} simultaneously or before the reception of S_{Slow2} by cell $k+1$ (cell k receiving the second signal S_{Slow} knows that it will never receives S_{Init}). When a cell enters state BK, it suppresses signal S_{Init}, sends to its right neighbor a new signal $S_{suppress}$ and suppresses the reflection of S_{Slow}. Its right neighbor receiving $S_{suppress}$ enters state L and emits S_L.

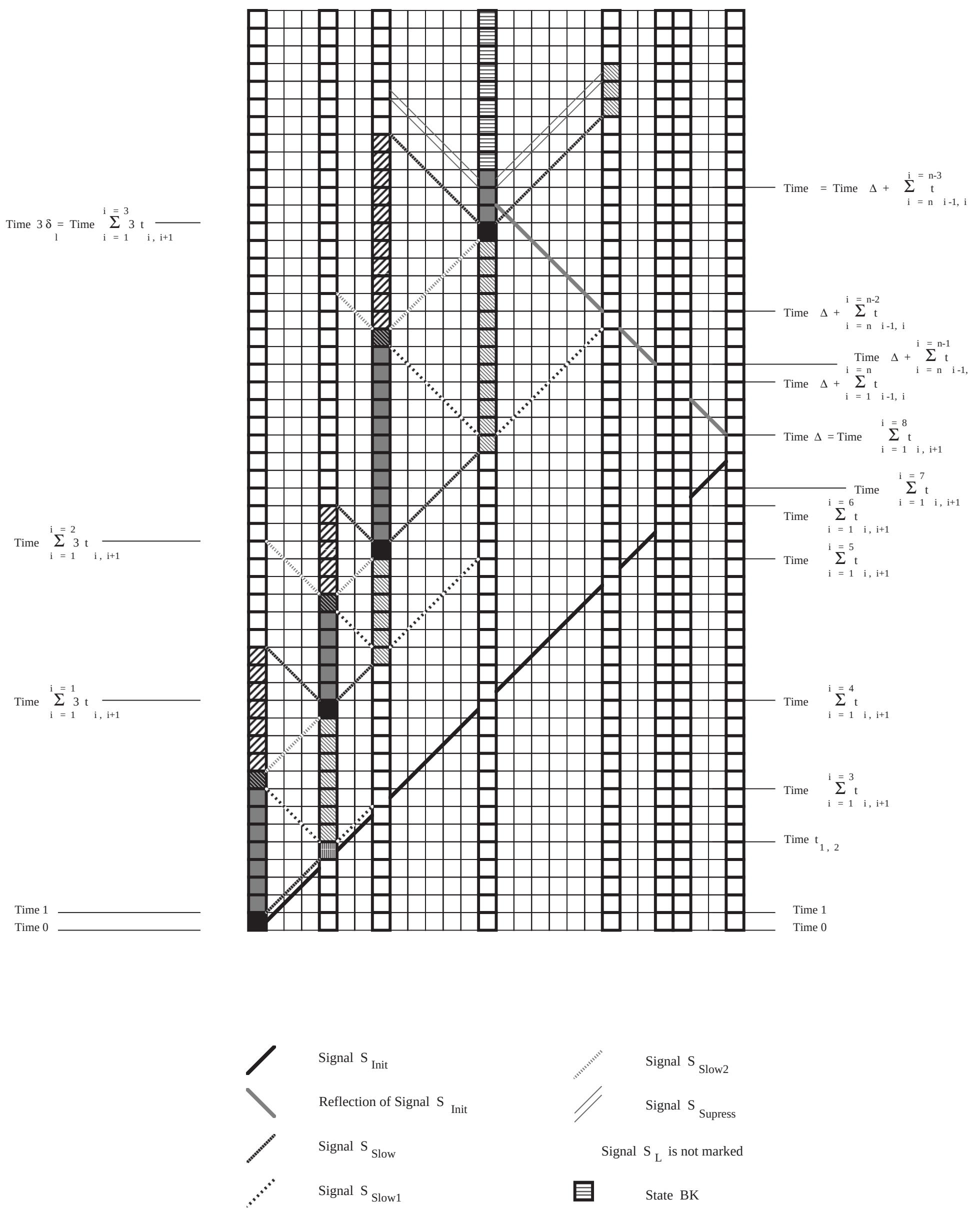


Figure 2: Breaking a nonuniform line in about its "middle"

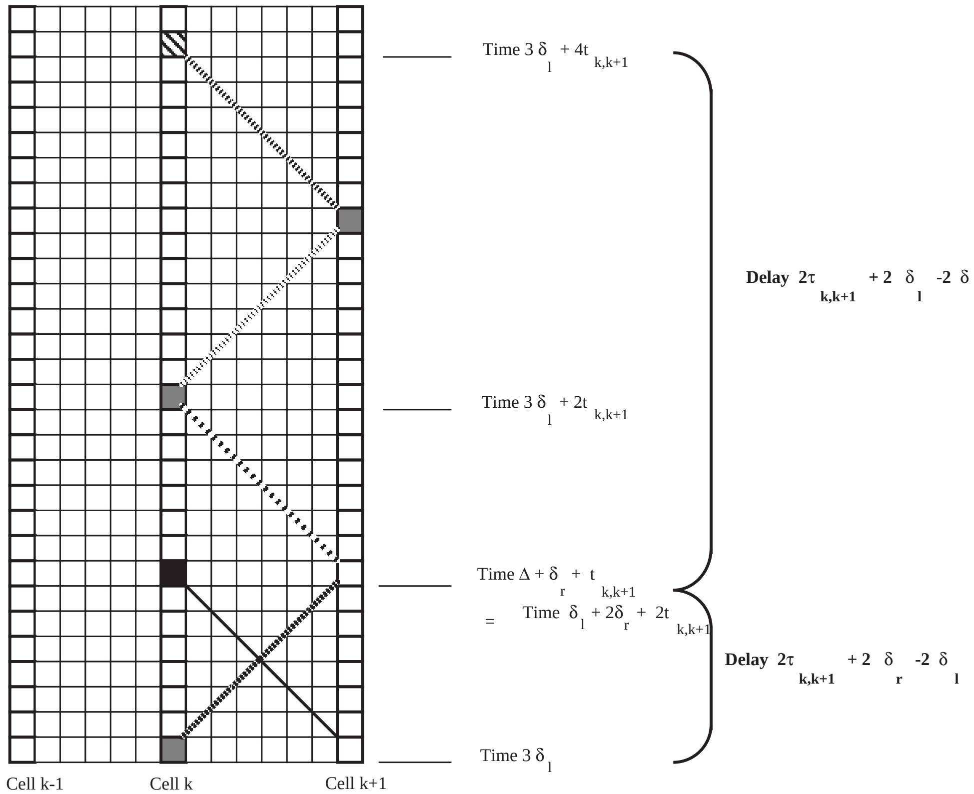


Figure 3: The involved delays in the process of the section 3

Thus, a cell k enters state BK at time $\Delta + \delta_{k,r} + t_{k,k+1}$; and we have $3\delta_{k,l} \leq \Delta + \delta_{k,r} + t_{k,k+1}$ (by the previous condition) and $3\delta_{k+1,l} > \Delta + \delta_{k,r}$ (cell $k+1$ does not enter state BK). This gives:
$3\delta_{k,l} \leq \Delta + \delta_{k,r} + t_{k,k1}$ and $3\left(\delta_{k,l} + t_{k,k+1}\right) > \Delta + \delta_{k,r}$;
$3\delta_{k,l} \leq 2\delta_{k,r} + \delta_{k,l} + 2t_{k,k+1}$ and $3\left(\delta_{k,l} + t_{k,k+1}\right) > 2\delta_{k,r} + t_{k,k+1} + \delta_{k,l}$;
$\delta_{k,l} \leq \delta_{k,r} + t_{k,k+1}$ and $\delta_{k,l} > \delta_{k,r} - t_{k,k+1}$;
thus, $|\delta_{k,r} - \delta_{k,l}| \leq t_{k,k+1}$.
But, by $\delta_{k,r} + t_{k,k+1} + \delta_{k,l} = \Delta$ and $\delta_{k,l} \leq \delta_{k,rl} + t_{k,k+1}$, we have:
$2\delta_{k,l} \leq \Delta \leq 2\left(\delta_{k,r} + t_{k,k+1}\right)$; it is to say;

$$\delta_{k,l} \leq \lfloor \frac{\Delta}{2} \rfloor \leq \lceil \frac{\Delta}{2} \rceil \leq \delta_{k,r} + t_{k,k+1}$$

We observe on the figure 2 that, in general, we do not have $\delta_{k,l} \leq \delta_{k,r}$. But the inequality 3 holds. If $\delta_{k,r} \leq \delta_{k,l}$, it is obvious; else the equality 2 ensures us that $0 \leq \delta_{k,r} - \delta_{k,l} \leq t_{k,k+1}$. □

Remark 1

The figure 3 describes the previous cut. We observe that if the line is cut on cell k:

- i - The delay between the arrival of S_{Slow} on cell k (which is $3\delta_{k,l}$) and the arrival of the reflection of S_{Init} (which is $\Delta + \delta_{k,r} + t_{k,k+1} = \delta_{k,l} + 2\delta_{k,r} + 2t_{k,k+1}$) is $2t_{k,k+1} + 2\delta_{k,r} - 2\delta_{k,l}$.

- ii - The delay between the arrival of S_{Init} (which is $\Delta + \delta_{k,r} + t_{k,k+1} = \delta_{k,l} + 2\delta_{k,r} + 2t_{k,k+1}$) and the arrival of the signal S_{Slow} (emitted by cell $k+1$) on cell k (which is $3\delta_{k,l} + 4t_{k,k+1}$) is $2t_{k,k+1} + 2\delta_{k,l} - 2\delta_{k,r}$.

Thus, if we sightly modify the previous algorithm: cell $k+1$ always emits the equivalent of S_{Slow} to cells k and $k+2$, at the break, cell k always knows the values of $2t_{k,k+1} + 2\delta_{k,l} - 2\delta_{k,r}$ and of $2t_{k,k+1} + 2\delta_{k,r} - 2\delta_{k,l}$ as delays between arrivals of signals.

In the following, we shall need the values of $2t_{k,k+1} + 2\delta_{k,l} - 2\delta_{k,r}$ and of $2t_{k,k+1} + 2\delta_{k,r} - 2\delta_{k,l}$. Thus, we add the two previous exchange of signals these new signals: when a cell receives signal S_{Slow} of its left neighbor, it sets up a loop between itself and its right neighbor (it sends signals, denoted by Σ which are reflected by itself and its right neighbor); and it always sends signals $\Sigma_{r,l}$ up to the possible reflection of S_{Slow} or S_{Init}. If this cell receives signal S_{Slow} of its right neighbor, it releases this loop (suppressing all signals $\Sigma_{r,l}$); else, at the reception of the reflection of S_{Init}, it sends signals $\Sigma_{r,l}$ up to the reception of the equivalent of S_{Slow} emitted by its right neighbor and maintains the loop. In other words, the loop is maintained if and only if the break occurs.

4 Synchronization of a two cells line

The synchronization of two finite automata was studied previously in [6]. Here, in order to obtain a synchronization of a (whole) line in quadratic time, we present a simple solution for two cells in quadratic time. In this section, cells of the line are cell 1 and cell 2 and $t_{1,2}$ is denoted by $\Delta - 1$.

The link between the two cells is two ways: we distinguish these ways. Cell 1 (cell 2) sends a signal to cell 2 (cell 1) using way 1 (way 2). By this way, we define a loop between our two cells. Thus, we can understand the time evolution of exchange of signals as occurring on a cylinder; but for convenience we represent this evolution on a "ribbon". When cell 1 enters state G, it emits a signal S_T (time signal) to cell 2 on way 1; cell 2 receives this signal after $\Delta - 1$ units of time and sends it to cell 1 after Δ units of time; cell 1 receives it after $2\Delta - 1$ units of time and sends it again after 2Δ units of time; and so on ... This signal S_T can be viewed as a clock which marks every 2Δ units of time. In the following, we shall always assume that signal S_T always come back and

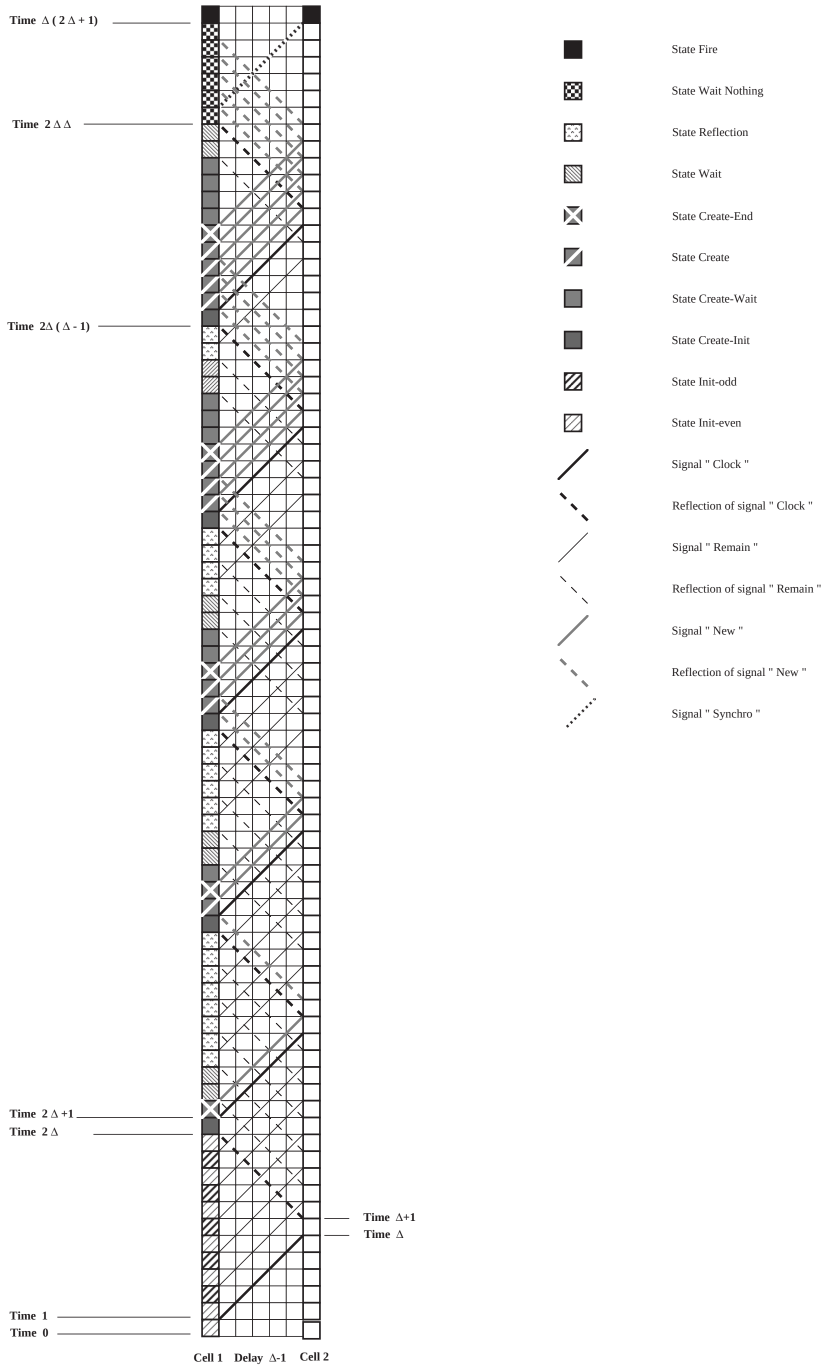


Figure 4: Synchronization of a two cells line

forth between the two cells. This allows us to speak of the p^{th} period of cell 1 between time $2(p-1)\Delta$ (included) and the time $2p\Delta$ (excluded).

Proposition 2 *There exists an automaton $\mathcal{K}$ which synchronizes all nonuniform line Λ of two automata in time $\Delta(2\Delta+1)$.*

Proof

The figure 4 illustrates this proof.

1. After the first emission of S_T, called signal "*Clock*", cell 1 emits a new signal "*Remain*" one unit of time out of two until it receives the reflection of the *Clock* signal. This needs only two states: $Init - Even$ and $Init - Odd$. By this way, $\Delta - 1$ signals *Remain* are emitted by cell 1 and then reflected by cell 2.

2. Then during the j^{th} period (with $1 < j \leq \Delta$), cell 1 emits $j-1$ successive signals "*New*" and reflects only $\Delta - 1 - j$ signals *Remain* (see point 3) above). Then, after the end of the Δ^{th} period (at time $2\Delta^2$), cell 1 sends to cell 2 a special signal "*Synchro*" and enters a new state "*Wait Nothing*". When cell 2 receives *Synchro* (at time $2\Delta^2 + \Delta$), it enters the Fire; and, when cell 1 in state *Wait Nothing* does not receive a reflected signal *New*, it enters the Fire (at time $2\Delta^2 + \Delta$).

3. Let us describe emission of *New* and suppression of signals *Remain*. When cell 1 receives reflection of *Clock*, it enters the state $Create - Init$. If in $Create - Init$ it receives the reflection of *New*, cell 1 enters state *Create* and emits signal *New*. As long as in *Create* cell 1 receives a reflected *New* signal, it remains in *Create* and emits *New*. If in *Create* or in $Create - init$ cell 1 does not receive a reflected signal *New*, it enters the state $Create - End$ and emits a signal *New* (thus, one more than in the last period). In $Create - End$ cell 1 enters the state $Create - Wait$ and emits the quiescent signal s_L. Cell 1 remains in $Create - Wait$ emitting s_L until it receives a reflected signal *Remain*. When in signal $Create - Wait$ cell 1 receives a reflected signal *Remain*, it enters state *Wait* and emits s_L (and thus suppresses the signal *Remain*). Cell 1 remains two units of time in *Wait* and when it receives a reflected *Remain*, it enters state *Reflection* and emits *Remain*. Finally, in *Reflection*, cell 1 only reflects reflections of *Remain* and remains in *Reflection* until the following period.

   We observe that, during a period, cell 1 is in state *Reflection* when it receives reflection of *Clock* if and only if it has emitted a *Remain* signal. If not (case of the Δ^{th} period) cell 1 receives reflection of *Clock* in state *Wait* and emits *Synchro* in order to achieve the process indicated in point 2).

This achieves the description of $\mathcal{K}$. □

Corollary 1 *For any natural α, there exists an automaton $\mathcal{K}$ which synchronizes every nonuniform line Λ of two automata in time $\theta(\Delta) = \Delta(2\Delta + 2\alpha + 1)$.*

Proof

Obviously, the two cells wait α periods until they enter the Fire. □

5 Moving and compressing data on loops

Let Λ be a nonuniform line of n automata $\mathcal{C}$ and two cells h and k, we define $\delta_{h,k}$ by:
if $h < k$ then

$$\delta_{h,k} = \sum_{i=h}^{i=k-1} t_{i,i+1}$$

else

$$\delta_{h,k} = \delta_{k,h}$$

The two following lemmas gives useful technic.

Lemma 1 *There exists an automaton which for any nonuniform line Λ of n such automata:*

1. *waits $2\delta_{h,k}$ on $Loop(h,k)$;*
2. *waits $\delta^2_{h,k}$ on $Loop(h,k)$;*
3. *waits $4\delta_{h,k}\delta_{h,m}$ on $Loop(h,k)$ and $Loop(h,m)$;*
4. *waits $4\delta_{h,k}\delta_{h,m} + \delta_{h,k}$ on $Loop(h,k)$ and $Loop(h,m)$.*

Proof

1. At time t, cell h emits a signal to cell k; this signal is reflected by cell k and thus, reaches cell $h$at time $t + 2\delta_{h,k}$.
2. A signal S_p goes back and forth between the cells h and k and thus it defines periods of $2\delta_{h,k}$ units of time. During a period, after reflecting (or emitting) a signal S_p, cell h puts in the loop ($Loop(h,k)$) four signals σ_{count} and one signal σ_{new}. These signals are always reflected up to the end of the $\delta^{th}_{h,k-1}$ period (which is easily known by some trick as in section 4). Two cases occur (see the figure 5):
   - if it remains only four units of time between emission by cell h of a signal σ_{count} and emission by cell h of S_p, then $\delta_{h,k}$ is even and the next coming back signal S_p is understood as S_{output}, indicating the end of its wait;

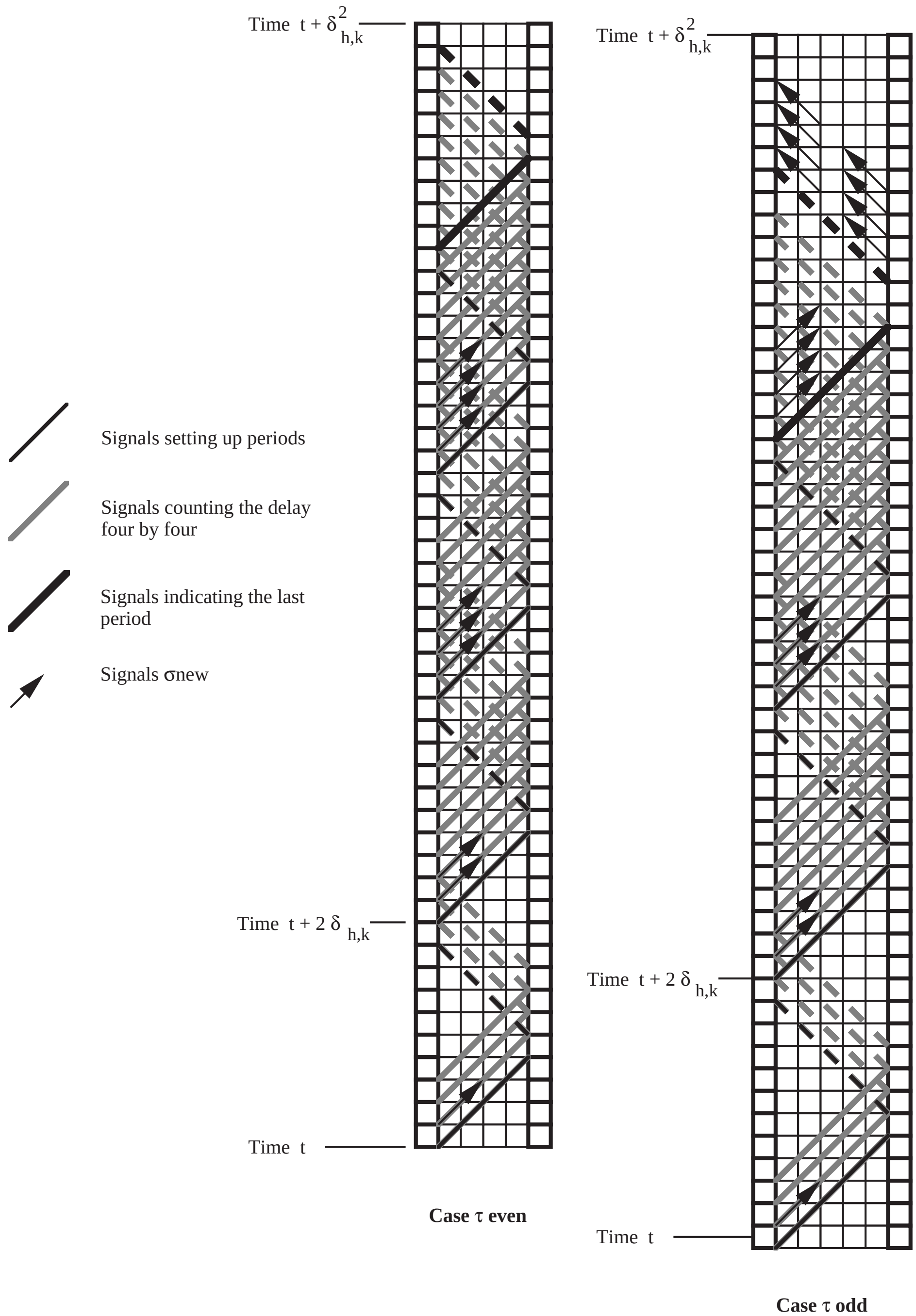


Figure 5: Proof of the lemma 1, point 2)

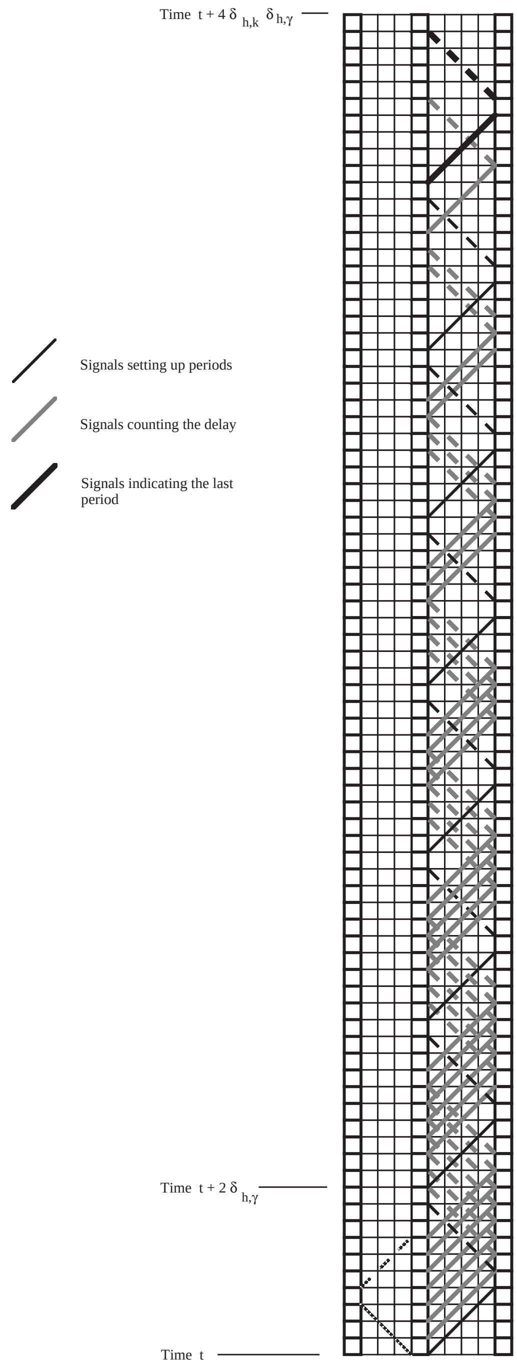


Figure 6: Proof of the lemma 1, point 3)

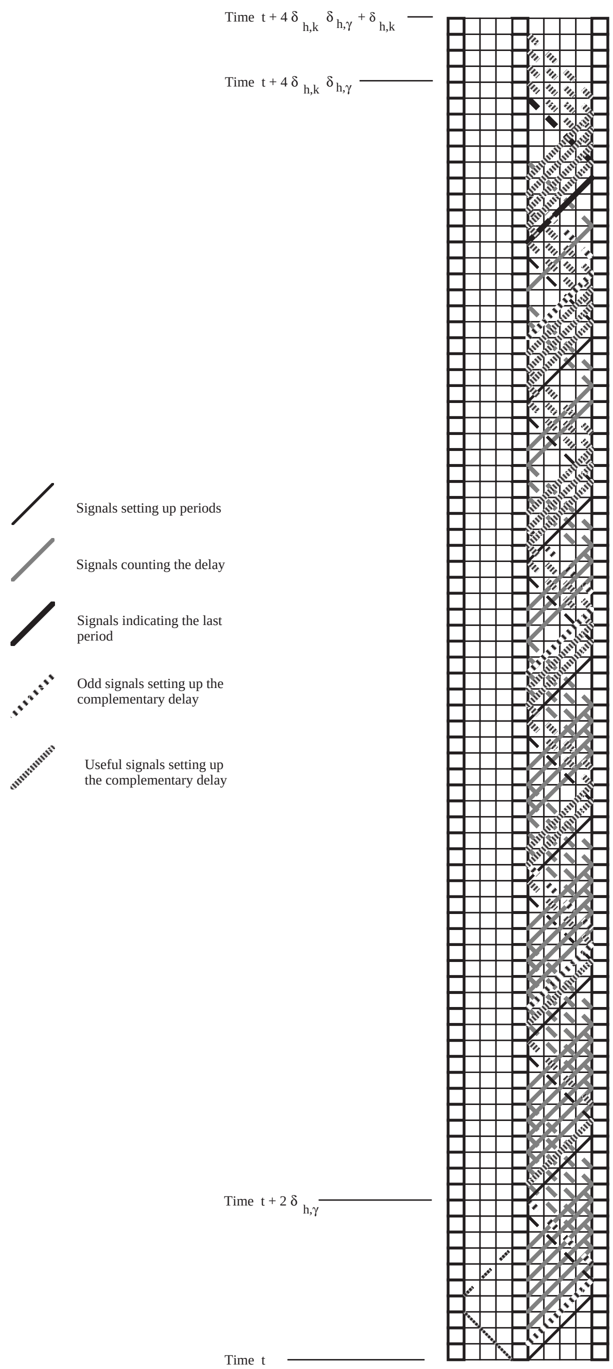


Figure 7: Proof of the lemma 1, point 4)

- if not, cell h knows that $\delta_{h,k}$ is odd. In this case, it has emitted $\delta_{h,k} - 1$. Thus, the last emitted signal σ_{new} reaches it at time $t + 2\delta_{h,k}(\delta_{h,k} - 1) + \delta_{h,k} - 1$ (which is $t + \delta_{h,k}^2 - 1$). Thus, cell h waits one unit of time and considers that it has received signal S_{output} at time $t + \delta_{h,k}^2$.

3. We suppose that $\delta_{h,k} \leq \delta_{h,m}$. As previously, cell h sets up periods of length $2\delta_{h,m}$ between itself and cell m. During the first period it sends a signal S to cell k and as long as the reflected signal S has not reached it, cell k sends signal S_1 to cell m. In $Loop(h,m)$, cell h counts periods of length $2\delta_{h,m}$ suppressing one signal S_1 by period as long as possible (see the figure 6).

4. As previously cell h waits $4\delta_{h,k}\delta_{h,m}$ units of time. At the beginning of a period on the loop $Loop(h,m)$, cell k adds a signal S_2 one period out of two and the last received signal S_2 is S_{output} (see the figure 7). □

Lemma 2 *There exists an automaton such that, whatever t and $\alpha \leq t_{k,k+1}$ are, receiving 2α consecutive signals from one of its neighbors between time t and time $t + t_{k,k+1}$, cell k emits exactly 2α successive signals from time $t + 2t_{k,k+1}^2$.*

Proof

- a - At time t, cell k emits a signal S_T; this signal is reflected by cell $k+1$ and cell k. This induces a loop between cells k and $k+1$ of period $2t_{k,k+1}$. If cell k marks all signals sent by cell $k+1$ by odd and even, it makes a loop of period $4t_{k,k+1}$. Now, we are sure that the 2α signals received by cell k are received during this new loop.

- b - The first point is now: "How to move 2α signals, appearing on a loop, during $\lfloor \frac{\alpha}{2} \rfloor$ periods ?". It is sufficient:

  - to create 4 new signals as soon as possible on the first period,
  - to suppress, if possible, 4 signals of the α's signals,
  - and, at the end of the $\lfloor \frac{\alpha}{2} \rfloor^{th}$ period to adjust the good number (its remainder $\pmod 4$)).

  The figure 8 illustrates such a process.

- c - The second point is now: "How to move and divide by 2 2α signals, appearing on a loop, during $\lfloor \frac{\alpha}{2} \rfloor$ periods ?". The process is the same than previously, adding 2 new signals and not 4. □

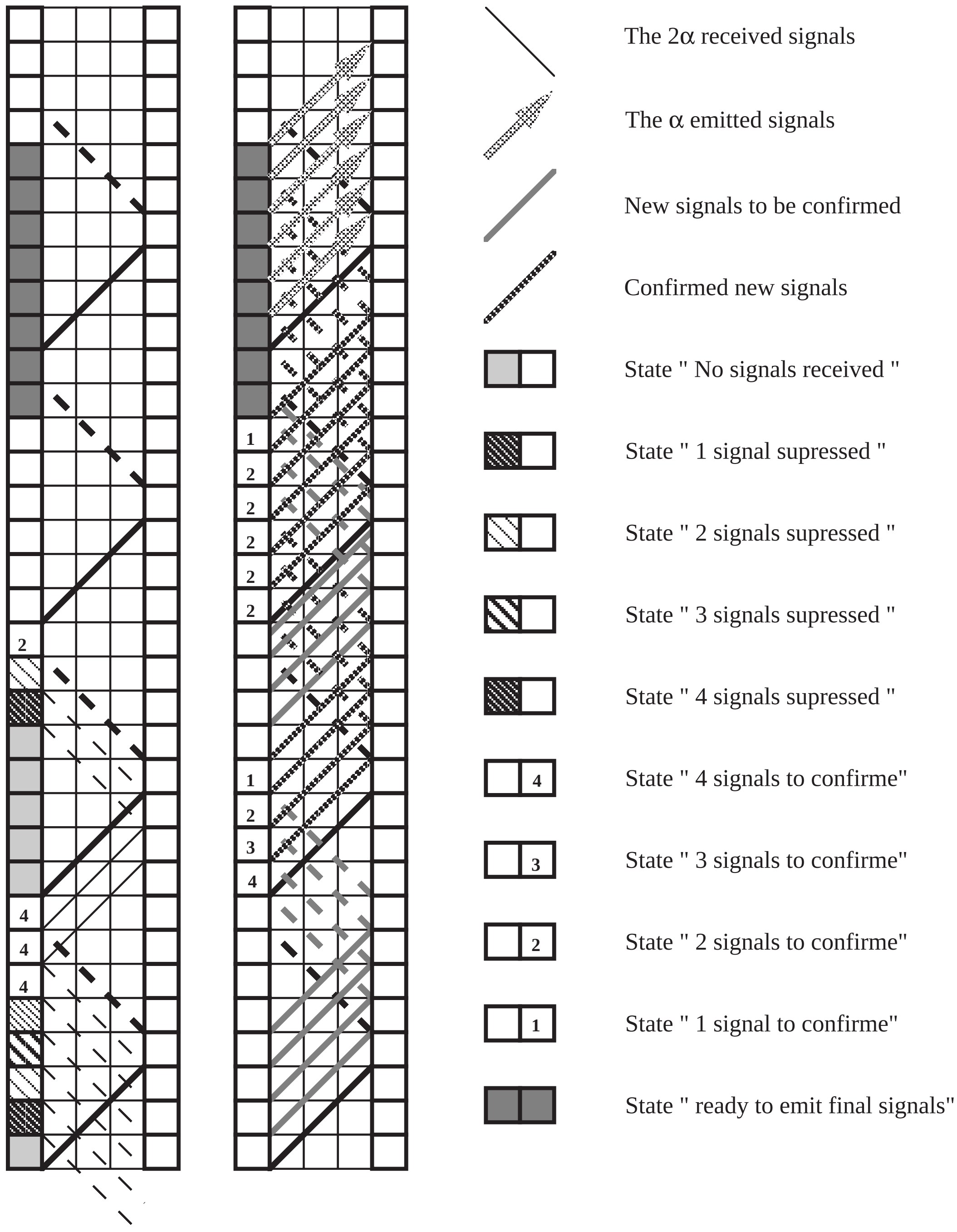


Figure 8: Proof of the lemma 2

6 Synchronization of a line

6.1

We choose to synchronize the whole line in a fixed time, say $\theta(\Lambda)$, we shall define later. When a cell, say cell k, enters state BK (of the proposition 1) at time $\Delta + \delta_{k,r} + t_{k,k+1}$, it waits Σ_r (Σ_l) units of time and emits a signal S_T ($S_T^\star$) in order to initialize the synchronization of the right (left) half line. But the synchronization of the right part will be achieved by cell $k+1$.Thus after time $\Delta + \delta_{k,r} + t_{k,k+1} + \Sigma_r + t_{k,k+1}$ (when cell $k+1$ receives the signal S_T emitted by cell k, cell $k+1$ acts on its right hand as the first cell of a new line (consisting of cells numbered from $k+1$ up to n): it is to say that it sends simultaneously S_{Init} (of the proposition 1) in order to break the right half line and the signal $Clock$ (of the proposition 2) in order to synchronize the half right line in time $\theta(t_{k,k+1})$ if this half line has only two cells (clearly the reflected $Clock$ indicates to cell k what case occurs). Second after time $\Delta + \delta_{k,r} + t_{k,k+1} + \Sigma_l$, cell k acts on its left hand cells numbered from k down to 1) as a general but interchanging the right and the left.

By this way, if our choice of Σ_r and Σ_l is such that synchronizations of the two half lines occur simultaneously at the expected time $\theta(\Delta)$, the whole line will be synchronized.

In order to achieve description of our automaton, we shall:

- i - define $\theta(\Delta)$,
- ii - compute Σ_r and Σ_l,
- iii - set up delays in such a way that setting up Σ_r (Σ_l) does not use the right half line, cells from 1 up to $k-1$ (the left half line, cells from $k+1$ up to n) after time $\Delta+\delta_{k,r}+t_{k,k+1}+\Sigma_l$ ($\Delta+\delta_{k,r}+2t_{k,k+1}+\Sigma_r$).

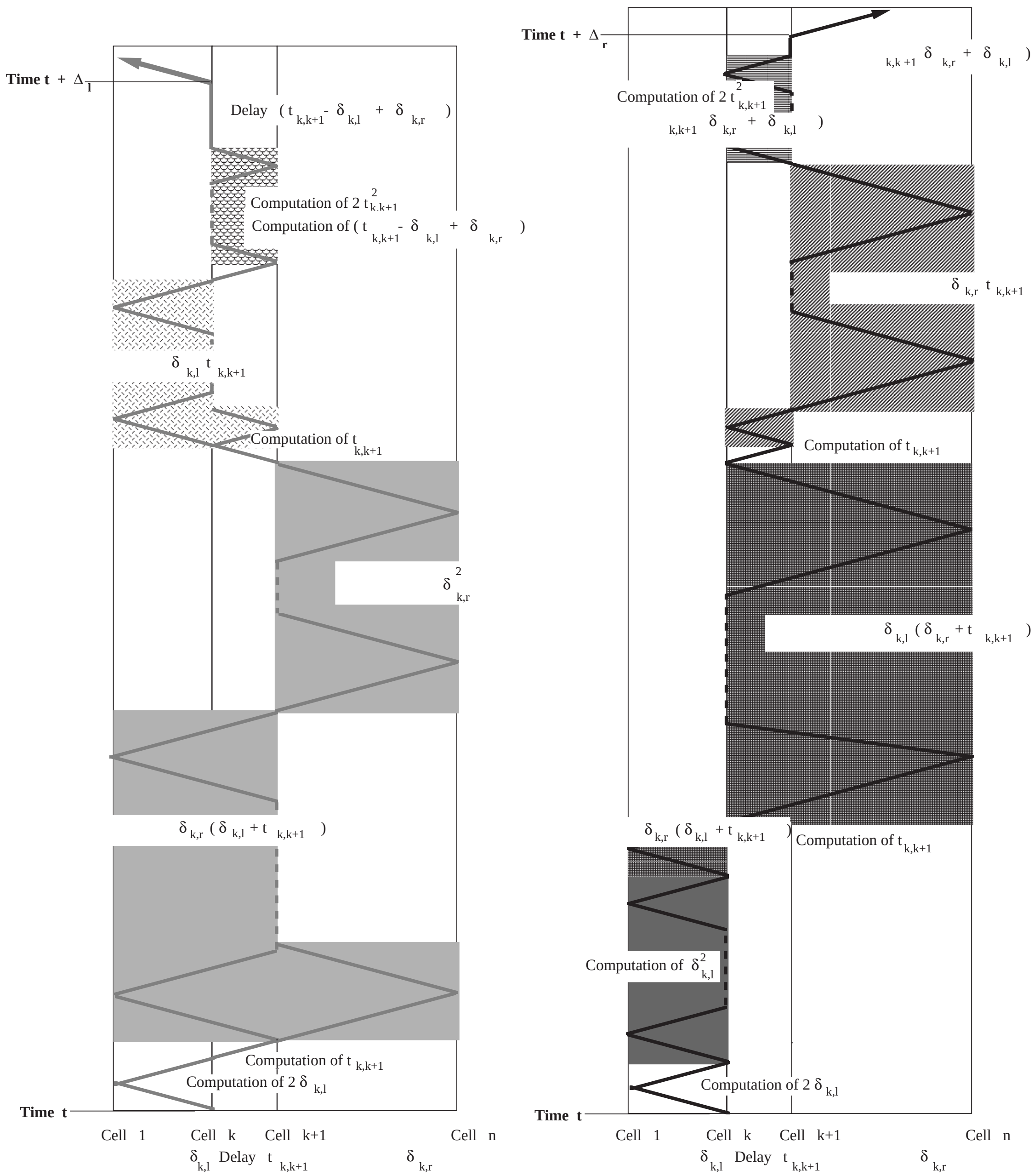
The point iii) is very important: if it is not realized, breaking the right part then the left part of the right part and so on can introduce an arbitrarily great number of signals which cannot be set up by finite automata.

6.2

We choose for $\theta(\Delta)$ the value previously obtained in the particular case of a line with only two cells and we fix the value of α later: $\Delta\,(2\Delta + (2\alpha+1))$.

Synchronization of the right (left) half line is obtained in $2\delta_{k,r}^2 + (2\alpha+1)\delta_{k,r}$ units of time ($2\delta_{k,l}^2 + (2\alpha+1)\delta_{k,r}$). Thus, these two synchronizations are achieved simultaneously at time $\Delta\,(2\Delta + (2\alpha+1))$ if and only if:
$\Delta + \delta_{k,r} + 2t_{k,k+1} + \Sigma_r + 2\delta_{k,r}^2 + (2\alpha+1)\delta_{k,r} = 2\Delta^2 + (2\alpha+1)\Delta$;
and $\Delta + \delta_{k,r} + 2t_{k,k+1} + \Sigma_l + 2\delta_{k,l}^2 + (2\alpha+1)\delta_{k,l} = 2\Delta^2 + (2\alpha+1)\Delta$.
We obtain

$$\Sigma_r = 2\Delta^2 + 2\alpha\Delta - 2t_{k,k+1} - 2\delta_{k,r}^2 - (2\alpha+2)\delta_{k,r}$$



Setting up the delay before the synchronization of the left half line

Setting up the delay before the synchronization of the right half line

Figure 9: Synchronization of a line

and

$$\Sigma_l = 2\Delta^2 + 2\alpha\Delta - t_{k,k+1} - 2\delta_{k,l}^2 - \delta_{k,r} - (2\alpha+1)\delta_{k,l}$$

But we have $\delta_{k,r}+t_{k,k+1}+\delta_{k,l} = \Delta$ (by the notation introduced at the beginning of the section 3), thus:
$\Sigma_r = 2\delta_{k,l}^2 + 4\delta_{k,l}\delta_{k,r} + 2t_{k,k+1}^2 + 4t_{k,k+1}(\delta_{k,r} + \delta_{k,l}) +$
$(2\alpha-2)t_{k,k+1} - 2\delta_{k,r} + 2\alpha\delta_{k,l}$ and
$\Sigma_l = 2\delta_{k,r}^2 + 4\delta_{k,l}\delta_{k,r} + 2t_{k,k+1}^2 + 4t_{k,k+1}(\delta_{k,r} + \delta_{k,l}) +$
$(2\alpha-1)t_{k,k+1} - (2\alpha-1)\delta_{k,r} - \delta_{k,l}$.

In order to set up easily the delays Σ_r and Σ_l, we choose the value of α in the corollary 1 such that $(2\alpha-2)t_{k,k+1} - 2\delta_{k,r} + 2\alpha\delta_{k,l}$ and $(2\alpha-1)t_{k,k+1} - (2\alpha-1)\delta_{k,r} - \delta_{k,l}$ are positive. By the proposition 1, point 1), $\alpha = 2$ is sufficient. We have:
$2t_{k,k+1} - 2\delta_{k,r} + 4\delta_{k,l} = 2\delta_{k,l} + 2(t_{k,k+1} - \delta_{k,r} + \delta_{k,l}) > 0$ and
$3t_{k,k+1} + 3\delta_{k,r} - \delta_{k,l} = 2\delta_{k,l} + 2t_{k,k+1} + 2(t_{k,k+1} - \delta_{k,r} + \delta_{k,l}) > 0$.
By our choice, we obtain:
$\Sigma_r = 2\delta_{k,l}^2 + 4\delta_{k,l}\delta_{k,r} + 2t_{k,k+1}^2 + 4t_{k,k+1}(\delta_{k,r} + \delta_{k,l}) +$
$2\delta_{k,l} + 2(t_{k,k+1} - \delta_{k,r} + \delta_{k,l})$ and
$\Sigma_l = 2\delta_{k,r}^2 + 4\delta_{k,l}\delta_{k,r} + 2t_{k,k+1}^2 + 4t_{k,k+1}(\delta_{k,r} + \delta_{k,l}) +$
$2\delta_{k,l} + 2t_{k,k+1} + (t_{k,k+1} + \delta_{k,r} - \delta_{k,l})$.
Thus, we have:
$\Sigma_r = 2\delta_{k,l} + 2t_{k,k+1}^2 + 2\delta_{k,l}^2 + 4\delta_{k,l}(\delta_{k,r} + t_{k,k+1})$
$+2(t_{k,k+1} - \delta_{k,r} + \delta_{k,l}) + 4t_{k,k+1}\delta_{k,r}$ and
$\Sigma_l = 2\delta_{k,l} + 2t_{k,k+1}^2 + 2\delta_{k,r}^2 + 4\delta_{k,r}(\delta_{k,l} + t_{k,k+1})$
$+(t_{k,k+1} + \delta_{k,r} - \delta_{k,l}) + 4t_{k,k+1}\delta_{k,l}$.
Finally, we write Σ_r and Σ_l as:
$\Sigma_r = 2\delta_{k,l} + 2\delta_{k,l}^2 + 4\delta_{k,l}(\delta_{k,r} + t_{k,k+1})$
$+4t_{k,k+1}\delta_{k,r} + 2t_{k,k+1}^2 + 2(t_{k,k+1} - \delta_{k,r} + \delta_{k,l})$ and
$\Sigma_l = 2\delta_{k,l} + t_{k,k+1} + 2\delta_{k,r}^2 + t_{k,k+1} + 4\delta_{k,r}(\delta_{k,l} + t_{k,k+1})$
$+4t_{k,k+1}\delta_{k,l} + 2t_{k,k+1}^2 + (t_{k,k+1} + \delta_{k,r} - \delta_{k,l})$.
thus, our choice of $\theta(\Delta)$ is $\Delta(2\Delta+5)$.

The figure 9 illustrates the following construction of Σ_r and Σ_l.

6.3

In this section, we explain how delay Σ_l can be set up.

- i - First, cell k waits $2\delta_{k,l}$ units on time on the $Loop(k,1)$ from time t up to time $t + 2\delta_{k,l}$ (see lemma 1, point 1)).

- ii - Then, cell k sends a signal to cell $k+1$. This signal reaches cell $k+1$ at time $t + 2\delta_{k,l} + t_{k,k+1}$.

- iii - Then (see lemma 1, point 3)), cell $k+1$ waits $4\delta_{k,r}(\delta_{k,l} + t_{k,k+1})$ units of time on the $Loop(k+1,1)$ and $Loop(k+1,n)$ from time $t + 2\delta_{k,l} + t_{k,k+1}$ up to time $t + 2\delta_{k,l} + t_{k,k+1} + 4\delta_{k,r}(\delta_{k,l} + t_{k,k+1})$.

- iv - Then (see lemma 1, point 2)), cell $k+1$ waits $4\delta_{k,r}^2$ units of time on the $Loop(k+1,n)$ from time $t+2\delta_{k,l}+t_{k,k+1}$ up to time $t+2\delta_{k,l}+t_{k,k+1}+4\delta_{k,r}(\delta_{k,l}+t_{k,k+1})$.

- v - Cell $k+1$ sends a signal to cell k. This signal reaches cell k at time $t+2\delta_{k,l}+2\delta_{k,r}^2+t_{k,k+1}+4\delta_{k,r}(\delta_{k,l}+t_{k,k+1})+t_{k,k+1}$.

- vi - Then (see lemma 1, point 3)), cell $k+1$ waits $4t_{k,k+1}\delta_{k,l}$ units of time on the $Loop(k,k+1)$ and $Loop(k,1)$ from time
$t+2\delta_{k,l}+2\delta_{k,r}^2+t_{k,k+1}+4\delta_{k,r}(\delta_{k,l}+t_{k,k+1})+t_{k,k+1}$
up to time
$t+2\delta_{k,l}+2\delta_{k,r}^2+t_{k,k+1}+4\delta_{k,r}(\delta_{k,l}+t_{k,k+1})+t_{k,k+1}+4t_{k,k+1}\delta_{k,l}$.

- vii - Then (see lemma 1, point 2)), cell $k+1$ waits $2t_{k,k+1}^2$ units of time on the $Loop(k,k+1)$ from time
$t+2\delta_{k,l}+2\delta_{k,r}^2+t_{k,k+1}+4\delta_{k,r}(\delta_{k,l}+t_{k,k+1})+t_{k,k+1}+4t_{k,k+1}\delta_{k,l}$
up to time
$t+2\delta_{k,l}+2\delta_{k,r}^2+t_{k,k+1}+4\delta_{k,r}(\delta_{k,l}+t_{k,k+1})+t_{k,k+1}+4t_{k,k+1}\delta_{k,l}+2t_{k,k+1}^2$.

- viii - During these $2t_{k,k+1}^2$ units of time on the $Loop(k,k+1)$, by lemma 2, cell k sets up the delay $t_{k,k+1}-\delta_{k,l}+\delta_{k,r}$ (observe that $2(t_{k,k+1}-\delta_{k,l}+\delta_{k,r})$ is known by the remark 1). By this way, it can send a signal at time
$t+2\delta_{k,l}+2\delta_{k,r}^2+t_{k,k+1}+4\delta_{k,r}(\delta_{k,l}+t_{k,k+1})+t_{k,k+1}+4t_{k,k+1}\delta_{k,l}+2t_{k,k+1}^2+(t_{k,k+1}-\delta_{k,l}+\delta_{k,r})$.
This time is $t+\Sigma_l$.

- ix - We observe that cells $k+1$ to n are released at time $t+2\delta_{k,l}+2\delta_{k,r}^2+t_{k,k+1}+4\delta_{k,r}(\delta_{k,l}+t_{k,k+1})$ (at the end of point - iv -).

- x - In order to show that the condition iii) of 6.2 is realized, we verify that:
$2\delta_{k,l}+2\delta_{k,r}^2+t_{k,k+1}+4\delta_{k,r}(\delta_{k,l}+t_{k,k+1})\leq\Sigma_r+t_{k,k+1}$ (recall that t is $\Delta+\delta_{k,r}+\delta_{k,l}$). It is to say:
$2\delta_{k,l}+2\delta_{k,r}^2+t_{k,k+1}+4\delta_{k,r}(\delta_{k,l}+t_{k,k+1})\leq 2\delta_{k,l}+2\delta_{k,l}^2+4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})+4t_{k,k+1}\delta_{k,r}+2t_{k,k+1}^2+2(t_{k,k+1}-\delta_{k,r}+\delta_{k,l})+t_{k,k+1}$. This can be written:
$2\delta_{k,r}^2\leq 2\delta_{k,l}^2+4t_{k,k+1}\delta_{k,l}+2t_{k,k+1}^2+2(t_{k,k+1}-\delta_{k,r}+\delta_{k,l})$ and
$2\delta_{k,r}^2\leq 2(\delta_{k,l}+t_{k,k+1})^2+2(t_{k,k+1}-\delta_{k,r}+\delta_{k,l})$.
But, we have $t_{k,k+1}+\delta_{k,l}\geq\delta_{k,r}$ by the proposition 1, point 3), and thus, the previous equality holds.

6.4

In this section, we do not explain how the delay Σ_r can be set up on cell $k+1$ but how the delay $\Sigma_r + t_{k,k+1}$ can be set up on cell k.

- i - First, cell k waits $2\delta_{k,l}$ units of time on the $Loop(k,1)$ (see lemma 1, point 1)).

- ii - Then (see lemma 1, point 2)), cell k waits $2\delta_{k,l}^2$ units of time on the $Loop(k,1)$ from time $t+2\delta_{k,l}$ up to time $t+2\delta_{k,l}+2\delta_{k,l}^2$.

- iii - Then (see lemma 1, point 3)), cell k waits $4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})$ units of time on the $Loop(k,1)$ and $Loop(k,n)$ from time $t+2\delta_{k,l}+2\delta_{k,l}^2$ up to time $t+2\delta_{k,l}+2\delta_{k,l}^2+4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})$.

- iv - Then, cell k sends a signal to cell $k+1$. This signal reaches cell $k+1$ at time $t+2\delta_{k,l}+2\delta_{k,l}^2+4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})+t_{k,k+1}$.

- v - Then (see lemma 1, point 3)), cell $k+1$ waits $4t_{k,k+1}\delta_{k,r}$ units of time on the $Loop(k,k+1)$ and $Loop(k+1,n)$ from time
$t+2\delta_{k,l}+2\delta_{k,l}^2+4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})+t_{k,k+1}$
up to time
$t+2\delta_{k,l}+2\delta_{k,l}^2+4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})+t_{k,k+1}+4t_{k,k+1}\delta_{k,r}$.

- vi - Then (see lemma 1, point 2)), cell $k+1$ waits $2t_{k,k+1}^2$ units of time on the $Loop(k+1,k)$ from time
$t+2\delta_{k,l}+2\delta_{k,l}^2+4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})+t_{k,k+1}+4t_{k,k+1}\delta_{k,r}$
up to time
$t+2\delta_{k,l}+2\delta_{k,l}^2+4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})+t_{k,k+1}+4t_{k,k+1}\delta_{k,r}+2t_{k,k+1}^2$.

- vii - During these $2t_{k,k+1}^2$ units of time on the $Loop(k,k+1)$, by the lemma 2, cell $k+1$ sets up the delay of $2(t_{k,k+1}-\delta_{k,r}+\delta_{k,l})$ (known by the remark 1). By this way, it can send a signal at time
$t+2\delta_{k,l}+2\delta_{k,l}^2+4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})+t_{k,k+1}+4t_{k,k+1}\delta_{k,r}+2t_{k,k+1}^2+2(t_{k,k+1}-\delta_{k,r}+\delta_{k,l})$. This time is $t+\Sigma_r+t_{k,k+1}$.

- viii - We observe that cells 1 to k are released at time $t+2\delta_{k,l}+2\delta_{k,l}^2+4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})$ (at the end of the point - iii -).

- ix - In order to show that the condition iii) of 6.2 is realized, we verify that:
$2\delta_{k,l}+2\delta_{k,l}^2+4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})\le\Sigma_l$ (recall that t is $\Delta+\delta_{k,r}+\delta_{k,l}$).
It is to say:
$2\delta_{k,l}+2\delta_{k,l}^2+4\delta_{k,l}(\delta_{k,r}+t_{k,k+1})\le$
$2\delta_{k,l}+t_{k,k+1}+4\delta_{k,r}(\delta_{k,l}+t_{k,k+1})+2\delta_{k,r}^2+t_{k,k+1}+4t_{k,k+1}\delta_{k,l}+2t_{k,k+1}^2+(t_{k,k+1}+\delta_{k,r}-\delta_{k,l})$. This can be written:
$\delta_{k,l}^2\le 2t_{k,k+1}+4\delta_{k,r}t_{k,k+1}+2\delta_{k,r}^2+2t_{k,k+1}2+(t_{k,k+1}+\delta_{k,r}-\delta_{k,l})$

$\delta_{k,l}^2 \leq 2(t_{k,k+1} + \delta_{k,r})^2 + 3t_{k,k+1} + \delta_{k,r} - \delta_{k,l})$.
But, we have $t_{k,k+1} + \delta_{k,l} \geq \delta_{k,r}$ by the proposition 1, point 3), and thus, the previous equality holds.

6.5

The previous construction allows us to state the following theorem:

Theorem 1 *There exists an automaton which synchronizes every nonuniform line Λ in time $\Delta(2\Delta + 5)$.*

7 Conclusion

In conclusion, it is possible to synchronize a nonuniform line using usual Minsky's strategy for the Firing Squad. The line is not broken on exactly its middle, but relations between delays insure us to achieve the synchronization in a quadratic time.

Clearly, if we define optimality in the sense of K. Kobayashi [4], there does not exists an optimal time solution (this is not the case even if the line has only two cells [6]). In [6], it has been proved that synchronization of two cells has not a linear solution; thus synchronization of a nonuniform line has not a linear solution. We do not know if there exist a synchronization of a nonuniform line in time $O(\Delta \log(\Delta))$. As think by T. Jiang [3], in some cases (a line) it is possible to synchronize a nonuniform network in a quadratic time.

This gives us a new strategy to synchronize any nonuniform graph of automata (using Rosenstiehl's method [8]), but, in this case, the synchronization time does not depend on the delayradius and is quadratic in the time needed to go through any cell (this time is given by the local orientation involved in the Rosenstiehl's algorithm).

References

[1] R. Balzer. *A 8 states minimal time solution to the Firing Squad Synchronization Problem*, Information and Control 10, pp 22-42 1967.

[2] T. Jiang. *The synchronization of Nonuniform Networks of Finite Automata*, 30^{th} Symposium on Foundations of Computer Science, pp 376-381 1989.

[3] T. Jiang. *The synchronization of Nonuniform Networks of Finite Automata*, Information and Communication 97, pp 234-261 1992.

[4] K. Kobayashi. *On the minimal time of the Firing Squad Synchronization Problem for Polyautomata Networks*, Theoretical Computer Science 7, pp 300-318 1978.

[5] J. Mazoyer. *A six states minimal time solution to the Firing Squad Synchronization Problem*, Theoretical Computer Science 50, pp 183-238 1987.

[6] J. Mazoyer. *Synchronization of two finite automata*, LIP Report 92-34 1992.

[7] M. Minsky. *Finite and infinite machines pp 28-29*, Prentice Hall,1967.

[8] P. Rosenstiehl, J.R. Fiksel and A. Holliger. *Intelligent graphs: networks of finite automata capable of solving graph problems*, Graph theory and Computing (R.C. Read Ed) Academic Press New-York, 1972.

[9] V. Varshavsky, V. Marakhovsky and A. Peschansky. *Synchronization of interacting automata*, MST 14, pp 212-230 1970.

[10] A. Waksman. *An optimal solution to the Firing Squad Synchronization Problem*, Information and Control 9, pp 66-87 1966.