

## GOTO'S CONSTRUCTION AND PASCAL'S TRIANGLE: NEW INSIGHTS INTO CELLULAR AUTOMATA SYNCHRONIZATION

JEAN-BAPTISTE YUNÈS <sup>1</sup>

<sup>1</sup> LIAFA, Université Paris Diderot/CNRS  
Case 7014  
75205 Paris Cedex 13. France.  
*E-mail address:* Jean-Baptiste.Yunes@liafa.jussieu.fr  
*URL:* <http://www.liafa.jussieu.fr/~yunes/>

---

ABSTRACT. Here we present a new non-recursive minimal-time solution to the Firing Squad Synchronization Problem which does not use any recursive process. In 1962, E. Goto designed an iterative algorithm which uses Minsky-McCarthy's solutions to synchronize in minimal-time. Our solution does not use any standard recursion process, only some "fractal computation", making it a purely iterative synchronization algorithm.

### Introduction

The firing squad synchronization problem (FSSP for short) has been the subject of many studies since 1957 when Myhill stated it and Moore reported it (see [Mo64]). We can state the problem as follows:

Does there exist a finite automaton such that a chain of  $n$  (whatever  $n$  could be) such automata would be synchronized at some time  $\mathcal{T}(n)$  after being initiated at time  $t = 0$ ? Each automaton is connected with its two neighbors and is assumed to be structurally independent of the number  $n$ . The synchronization is a configuration such that each automaton is in a so-called firing state which was never used before time  $\mathcal{T}(n)$  and the ignition configuration is a configuration such that every automaton but the first one of the chain is in a quiescent state.

Besides the fact that numerous papers were published about it and many different solutions were designed to solve the problem in various conditions, one of the very first solution made by Goto remained mythical for a long time. His courses notes are not available and Goto has not published his solution elsewhere. Many years later, Umeo (see [Um96]) was the first who tried to reconstruct it as he was able to talk to Goto himself who then gave him some old incomplete drawing. After that, Mazoyer (see [Ma98]) made a possible reconstruction of it but did not published it.

---

*Key words and phrases:* parallel computations, synchronization, firing squad, cellular automata.

In this paper we do not try to strictly reconstruct Goto's solution but to use his main idea to build a new minimal-time solution with the following interesting characteristics in mind:

- the iterative process ensures that we do not need a complex discretization process;
- the set of signals used is small; we will see that an implementation will give us the opportunity to use only two different signals (slope 1 and slope 3);
- there is only one "cut" of the line;
- we want to obtain a solution whose energy consumption is lower than  $n^2$ ;

## 1. The schema

Figures 4, 5, 6 illustrate the overall mechanism involved. Roughly speaking, we can say that the main idea, due to Goto (see [Go62]) is to split the line into successive sublimes the lengths of which are a sequence of powers of 2 ( $2^0, 2^1, 2^2$ , etc.), from the left and from the right. We will use this iterative decomposition to place some minimal-time firing squad. Of course it is possible to use any minimal-time solution, even the one we are constructing as Gerken did, but we choose to use some specific solutions able to efficiently synchronize powers of 2 - see [Yu08].

As it is not always possible to cover the line with such left and right sublimes something must be built in the middle to ensure that the residue is also correctly synchronized, either ensuring some overlapping or filling the hole.

Different constructions are involved:

- splitting the line into successive sublimes of length powers of 2 from the left;
- starting appropriate minimal-time solutions on the left sublimes;
- splitting the line into successive sublimes of length powers of 2 from the right;
- starting appropriate minimal-time solutions on the right sublimes;
- filling the *empty space* in the middle of the line when necessary.

In the following we will always consider that cells of the chain are numbered from 0 (the left cell) to  $n - 1$  (the right cell) and that the time starts at 0.

## 2. Splitting the line into successive sublimes of length $2^i$ from the left

Splitting a line into successive power of 2 is easy and is illustrated in Figure 1(a). Suppose that length  $l$  has already been constructed in space, *i.e.*, the distance between the abscissas of the two sites  $P'_i$  and  $P'_{i-1}$  is  $l$ .

From  $P'_i$  two signals are issued. The first, of slope 1, goes to the left until it meets the previous stationary signal issued from  $P'_{i-1}$ , then bounces back in the reverse direction until it meets the signal of slope 2 issued from  $P'_i$  to the right. That crossing point  $P'_{i+1}$  is the new starting point of the next construction. If we start with  $l = 1$ , then it is easy to see that all powers of 2 are successively constructed in space.

The previous construction is due to Goto, but for different reasons that we will explain later, we need to also use another construction illustrated in Figure 1(b). Suppose that length  $l$  has been constructed and that  $P_i$  and  $P'_i$  are located on the same vertical and distant by  $l$ . Then, the meeting of the signal of slope 1 issued from  $P'_i$  and the signal of slope  $\frac{3}{2}$  issued from  $P_i$  is  $P_{i+1}$  and is exactly at distance  $2l$  (in space) from  $P_i$ . At the same time if we start a signal of slope 2 from  $P'_i$ , it meets the stationary signal issued from  $P_{i+1}$

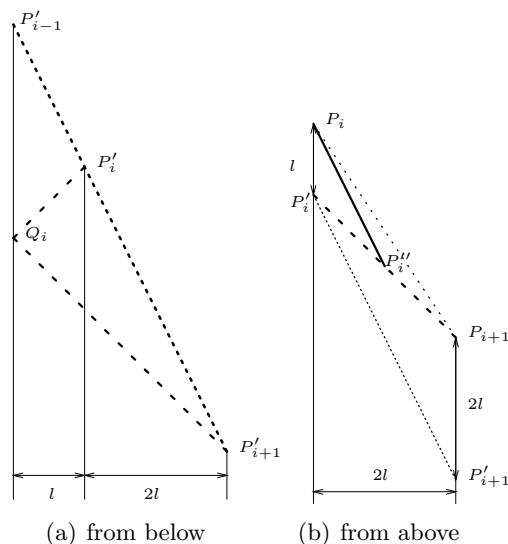


Figure 1: Constructing  $P_i$ 's,  $P'_i$ 's

at point  $P'_{i+1}$  distant from  $P_{i+1}$  by  $2l$ . Then again it is easy to see that if we start with  $l = 1$ , all powers of 2 are successively constructed in space.

We shall also need the middle  $P''_i$  of  $P'_i P_{i+1}$  which is obtained by starting a signal of slope 2 from  $P_i$ .

Starting from  $P'_1 = [1,1]$  then, with the preceding constructions, one can see that, for all  $i \geq 2$ , the constructed points have the following coordinates:

$$P_i = [2^i - 1, 3 \cdot (2^{i-1} - 1)] \tag{2.1}$$

$$P'_i = [2^i - 1, 2^{i+1} - 3] \tag{2.2}$$

$$P''_i = [3 \cdot 2^{i-1} - 1, 5 \cdot 2^{i-1} - 3] \tag{2.3}$$

$$Q_i = [2^{i-1} - 1, 5 \cdot 2^{i-1} - 3] \tag{2.4}$$

The previous signals and the above equations are more easily viewed with the help of Pascal's triangle modulo 2 (which is obtained via Wolfram's rule 60) as one can see in Figure 2.

### 3. Synchronizing left sublimes

To synchronize the cells of the left constructed sublimes, one can use the schema illustrated by Figure 3(a).

Every stationary signal issued from  $P_i$  meets the return of the main signal (a line of equation  $y = 2n - x$ ) at  $S_i = [2^i - 1, 2n - 2^i + 1]$  where a minimal-time solution can be started with an initiator at right using the stationary signal issued from  $S_{i-1}$  as a border.

### 4. Splitting the line into successive sublimes of length $2^i$ from the right and synchronizing them

The construction is illustrated by Figure 3(b). As one can see, we use the left construction to build the right sublimes. From every  $Q_i$  a signal of slope 1 is started to the right

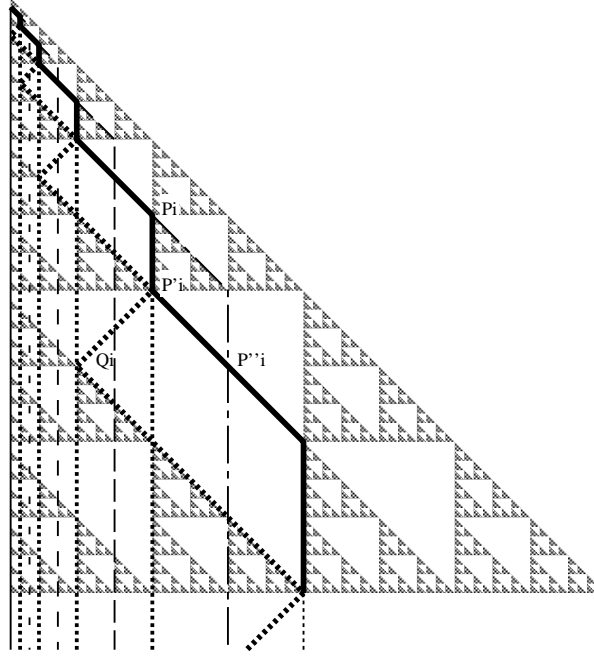


Figure 2: Rule 60 helps

until it meets the return of the main signal, then at that point a stationary signal is set up which will meet the symmetric counterpart of the return of the main signal issued from the middle of the line at a point named  $S'_i$ . From each of those points, a minimal-time solution can be started with an initiator at left, using the stationary signal issued from  $S'_{i-1}$  as a border.

## 5. Filling the empty space

Depending on the length of the line it is necessary to carefully consider what happens in the middle of the line. Points  $S_i$  and  $S'_i$  are symmetric relative to the middle of the line. Depending whether  $S_i$  is left or right to  $S'_i$ , the last left and right sublimes overlap or not. A simple analysis shows that there exists three different cases to consider about the position of points  $P$ ,  $P'$ ,  $P''$  relatively to the return of the main signal (signal of slope 1 issued from the left cell which bounces back from the right border).

Whatever be  $n$ , there exists  $i$  such that one and only one of the following cases occurs:

- (1)  $P_i$ ,  $P'_i$  and  $P''_i$  are all constructed before the main signal has returned and  $P_{i+1}$  is not;
- (2)  $P_i$  and  $P'_i$  are constructed before the main signal has returned and  $P''_i$  is not;
- (3)  $P_i$  is constructed before the main signal has returned and  $P'_i$  is not.

### 5.1. Case 1: P, P' and P'' before

Figure 4 illustrates the case where there is an index  $i$  such that  $P''_i$  has been built before the main signal has returned and it is not the case for  $P_{i+1}$ . With the help of Eqs 2.1

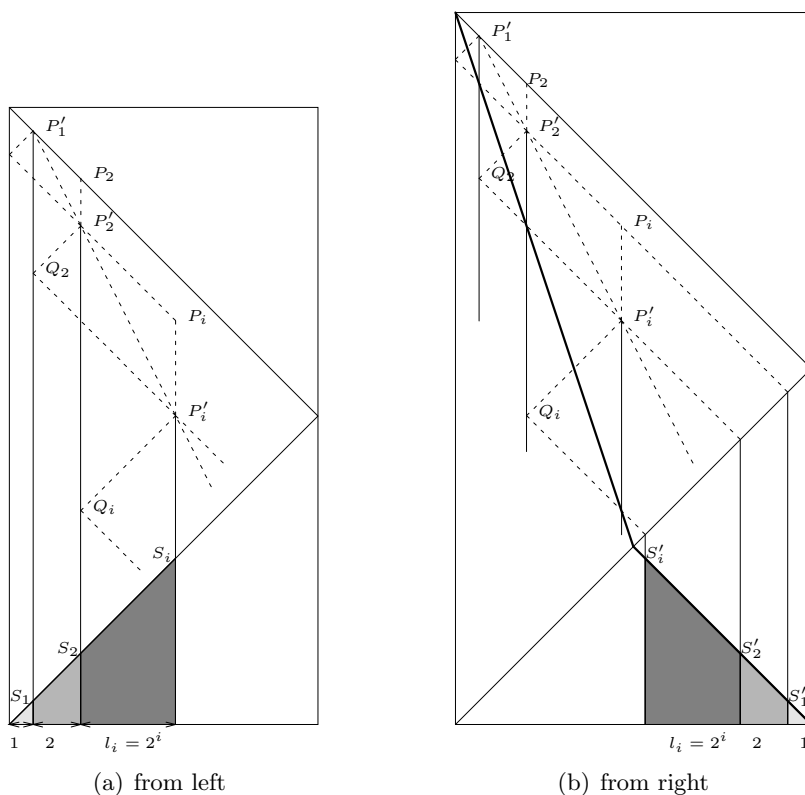


Figure 3: Synchronizing sublimes

and 2.3, these two conditions lead to:

$$\begin{cases} 5 \cdot 2^{i-1} - 3 \leq 2n - 3 \cdot 2^{i-1} + 1 \\ 2n - 2^{i+1} + 1 < 3 \cdot (2^i - 1) \end{cases}$$

then to

$$\begin{cases} 2^{i+1} - 2 \leq n \\ n < 2^{i+1} + 2^{i-1} - 2 \end{cases}$$

and the following equation holds

$$2^{i+1} - 2 \leq n < 2^{i+1} + 2^{i-1} - 2 \tag{5.1}$$

In that case synchronization is achieved by constructing the following firing squads:

- $i$  minimal-time FS are started on sites  $S_k = [2^k - 1, 2n - 2^k - 1]$  ( $1 \leq k \leq i$ );
- $i$  minimal-time FS are started on sites  $S'_k = [n - 2^k - 1, 2n - 2^k - 1]$  ( $1 \leq k \leq i$ ). A special case has to be considered for  $S'_i$  as  $P'_{i+1}$  has not been built by definition. To build  $S'_i$  it is sufficient to compute  $Q_i = [2^{i-1} - 1, 5 \cdot 2^{i-1} - 3]$  and its meeting point with the return of main signal, *i.e.* the meeting point of  $L : y = x + 2^{i+1} - 2$  and  $L' : y = 2n - x$  which has abscissa  $n - 2^i + 1$ ;
- an additional minimal-time FS can be started on site  $S_i$ , propagates to the right and uses the stationary signal issued from  $P''_i$  as its end-of-line.

For this schema to work we must verify that:

- $S'_i$  can always be built on time (it must appear before the middle of the line)

- $S'_i$  must appear at the left of  $P''_i$ , so that all cells in between  $S_i$  and  $S'_i$  can be synchronized by an appropriate FS.

We know that  $S'_i$  has coordinates  $n - 2^i + 1$  and that  $P''_i$  has abscissa  $5 \cdot 2^{i-1} - 3$  then we must have:

$$\begin{cases} \frac{n}{2} \leq n - 2^i + 1 \\ n - 2^i + 1 < 5 \cdot 2^{i-1} - 3 \end{cases}$$

which is

$$\begin{cases} 2^{i+1} - 2 \leq n \\ n < 7 \cdot 2^{i-1} - 4 \end{cases}$$

which is implied by Equation 5.1

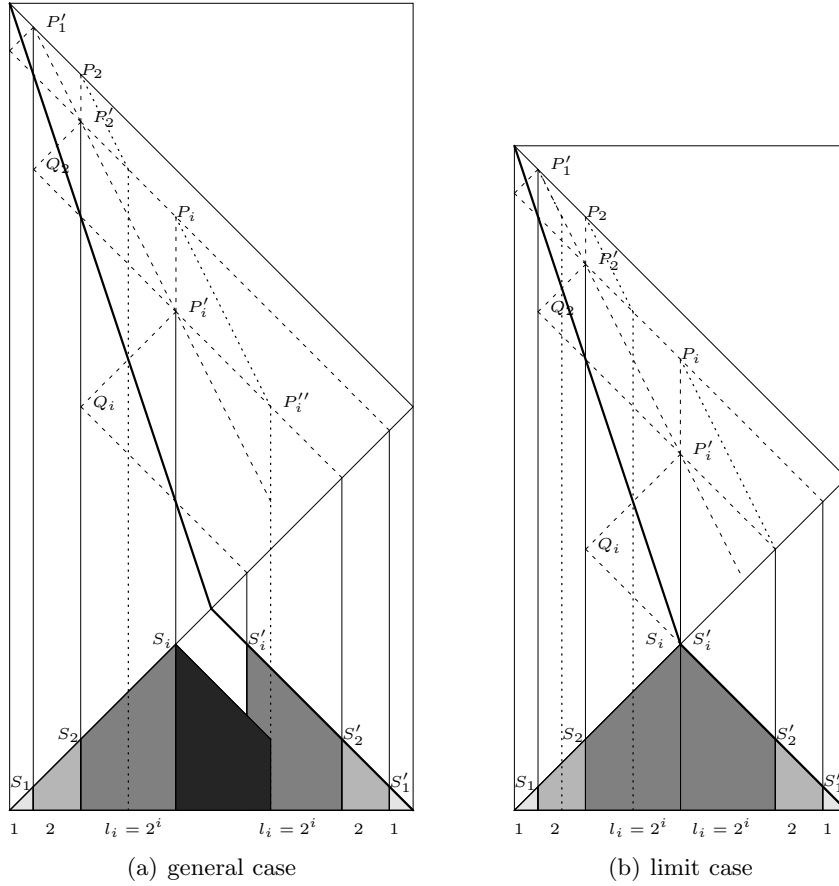


Figure 4: P, P' and P'' before

### 5.2. Case 2: P and P' strictly before, P'' after

Figure 5 shows what is constructed when there is an index  $i$  such that  $P'_i$  has been built strictly before the return of the main signal and that it is not the case for  $P''_i$ . With the help of Equations 2.2 and 2.3, these conditions lead to:

$$\begin{cases} 2^{i+1} - 3 < 2n - 2^i + 1 \\ 2n - 3 \cdot 2^{i-1} + 1 < 5 \cdot 2^{i-1} - 3 \end{cases}$$

which gives

$$\begin{cases} 3 \cdot 2^{i-1} - 2 < n \\ n < 2^{i+1} - 2 \end{cases}$$

and the following equation holds

$$3 \cdot 2^{i-1} - 2 < n < 2^{i+1} - 2 \tag{5.2}$$

In that case synchronization is achieved by the following constructions:

- $i$  minimal-time FS are started on sites  $S_k = [2^k - 1, 2n - 2^k - 1]$  ( $1 \leq k \leq i$ );
- $i - 1$  minimal-time FS are started on sites  $S'_k = [n - 2^k - 1, 2n - 2^k - 1]$  ( $1 \leq k < i$ );
- a minimal-time FS is started from  $S_i$  propagating to the right and using the stationary signal issued from  $P''_i$  as its end-of-line.

For this schema to work some conditions must be verified:

- $P''_i$  must be constructible;
- $S'_{i-1}$  must be at the left of  $P''_i$  so that the appropriate FS started at site  $S_i$  synchronizes at least the cells in between  $S_i$  and  $S'_{i-1}$ .

We know that the abscissa  $x$  of  $S'_{i-1}$  is solution of  $2n - x = x - 2^i + 1 + 2^{i+1} - 3$ , so that  $x = n - 2^{i-1} + 2$ . So we must have:

$$\begin{cases} 3 \cdot 2^{i-1} - 1 \leq n \\ n - 2^{i-1} + 2 \leq 3 \cdot 2^{i-1} - 1 \end{cases}$$

which gives

$$\begin{cases} 3 \cdot 2^{i-1} - 1 \leq n \\ n \leq 2^{i+1} - 3 \end{cases}$$

which is exactly the condition of Equation 5.2.

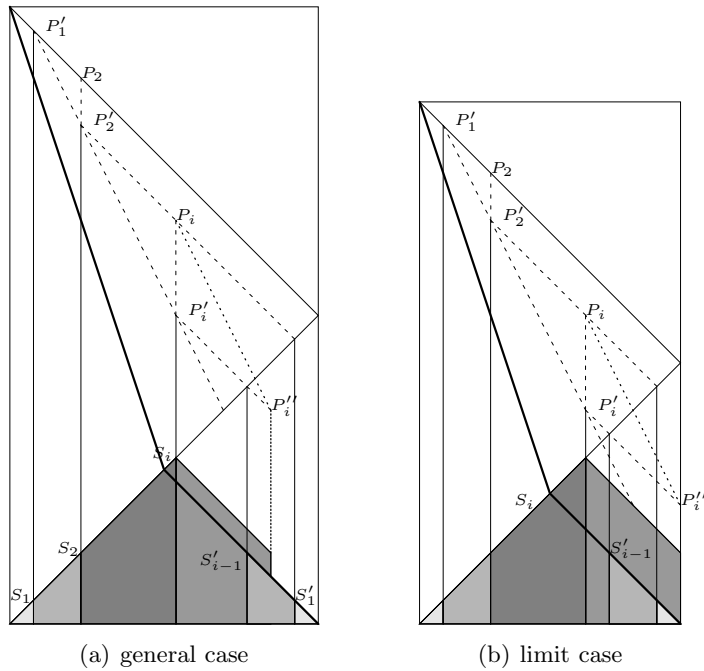


Figure 5: P, P' before, P'' after

### 5.3. Case 3: P before, P' and P'' after

Figure 6 illustrates the construction when there is an index  $i$  such that  $P_i$  has been built before the return of the main signal and that it is not the case for  $P'_i$ . Then with the help of Equations 2.1 and 2.2, we have:

$$\begin{cases} 3.(2^{i-1} - 1) \leq 2n - 2^i + 1 \\ 2n - 2^i + 1 \leq 2^{i+1} - 3 \end{cases}$$

thus

$$\begin{cases} 2^i + 2^{i-2} - 2 \leq n \\ n \leq 3.2^{i-1} - 2 \end{cases}$$

and the following equation holds

$$2^i + 2^{i-2} - 2 \leq n \leq 3.2^{i-1} - 2 \quad (5.3)$$

In that case synchronization is achieved by constructing the following firing squads:

- $i$  minimal-time FS are started on  $S_k = [2^k - 1, 2n - 2^k - 1]$  ( $1 \leq k \leq i$ );
- $i - 1$  minimal-time FS can be started on  $S'_k = [n - 2^k - 1, 2n - 2^k - 1]$  ( $1 \leq k < i$ ).  
Note that  $S'_{i-1}$  is built by a special process issued from  $Q_{i-1}$  as done in case 1.

For all this to work correctly, some conditions must be verified:

- $S'_{i-1}$  must appear on the left of  $S_i$ , such that an every cell will be synchronized.
- $S'_{i-1}$  must appear on time (before the middle cut of the line).

We know that the abscissa  $x$  of  $S'_{i-1}$  is solution of  $2n - x = x - 2^{i-2} + 1 + 5.2^{i-2} - 3$  then that  $x = n - 2^{i-1} + 1$ . So we must have:

$$\begin{cases} n - 2^{i-1} + 1 \leq 2^i - 1 \\ \frac{n}{2} \leq n - 2^{i-1} + 1 \end{cases}$$

which gives

$$\begin{cases} n \leq 3.2^{i-1} - 2 \\ 2^i - 2 \leq n \end{cases}$$

which is implied by Equation 5.3.

As for every integer  $n$ , there exists  $i$  such that  $2^i - 2 \leq n < 2^{i+1} - 2$ , from Equations 5.1, 5.2, 5.3, this proves the main result of this paper.

**Theorem 5.1** (Yunès). *The schema synchronizes every line of length  $n \in \mathbb{N}$ .*

## 6. Conclusion

A strict implementation of the preceding schema is possible but we would like to show how many interesting optimizations can be done.

First we can remark that if any minimal-time solution can be used to synchronize the sublimes, every subline has a length which is a power of 2. Then according to Yunès and Umeo (see [Yu08] and [Um07]), we know that it is possible to synchronize a line of length  $2^k$  with only 4 states, such solutions are algebraic and do not use any signal. Thus using such a construction will certainly lower down the total number of states.

But more than this, if we use one of these 4-state solutions then we can use them as the support for the construction of all the interesting points  $P$ ,  $P'$ ,  $P''$  and  $Q$  as one can see in Figure 2.

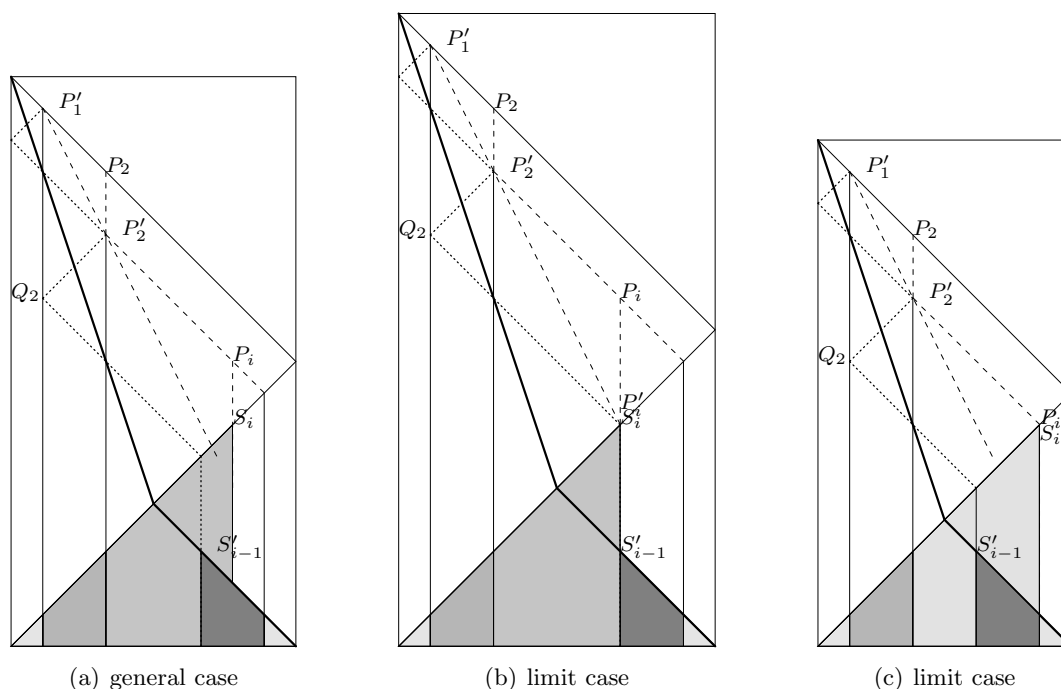


Figure 6: P before, P' and P'' after

Now one can remark that there are only two kind of signals: slope 1 and slope 3. And the signal of slope 3 is only used to cut the main line into two equals parts. We actually do not know if an explicit construction of this signal is necessary.

Besides the fact that such a schema is the very first one, one can observe that it has many interesting characteristics which probably nobody never thought about.

## References

- [Ge87] Hans-D. Gerken. *Über Synchronizations - Probleme bei Zellularautomaten*. Diplomarbeit, Institut für Theoretische Informatik, Technische Universität Braunschweig, **1987**.
- [Go62] Eiichi Goto. *A Minimum Time Solution of the Firing Squad Problem*. Courses Notes for Applied Mathematics 298, Havard University, pp. 52–59, **1962**.
- [Ma98] Jacques Mazoyer. *A minimal-time solution to the FSSP without recursive call to itself and with bounded slope of signals*. Unpublished draft, private communication, **1998**.
- [Mo64] Edward E. Moore. *Sequential Machines, Selected papers*. Addison-Wesley, **1964**.
- [Um96] Hiroshi Umeo. *A Note on Firing Squad Synchronization Algorithms*. IFIP Cellular Automata Workshop 96, Schloss Rauischholzhausen, Giessen, pp. 65, **1996**.
- [Um02] Hiroshi Umeo, Masaya Hisakoa, Takashi Sogabe. *An Investigation into Transition Rule Sets for Optimum-time Firing Squad Synchronization Algorithms on One-Dimensional Cellular Automata*. Interdisciplinary Information Sciences, Vol. 8, No. 2, pp. 207–217, **2002**.
- [Um07] Hiroshi Umeo, Naoki Kamikawa. *A 4-state solution to the firing squad based on Wolfram's rule 150*. Private communication, **2007**.
- [Yu08] Jean-Baptiste Yunès. *A 4-states Algebraic Solution to Linear Cellular Automata Synchronization*. Information Processing Letters. To appear, **2008**. doi: 10.1016/j.ipl.2008.01.009.