On the bound of the synchronization delay of a local automaton

Marie-Pierre Béal
Institut Gaspard Monge,
Université Paris 7 et CNRS

Jean Senellart
Laboratoire d'Automatique
Documentaire et Linguistique

October 30, 1997

Abstract

The synchronization delay of an N-state local automaton is known to be $O(N^2)$. It has been conjectured by S. Kim, R. McNaughton and R. McCloskey that, for deterministic local automata, it is $O(N^{1.5})$ on a two-letter alphabet and no less than $O(N^2)$ in the general case. We prove that this conjecture is false and that the synchronization delay is $\Omega(N^2)$ in all cases.

1 Introduction

Local automata are finite automata with a very strong synchronizing property: there are integers k and d $(0 \leq d \leq k)$, such that two paths of length k with the same label go through the same state at time d. The smallest integer k satisfying this property is called the synchronizing delay of the local automaton. Local automata recognize strictly locally testable languages of finite words, that is languages L on an alphabet A with $L - \{\epsilon\} = UA^* \cap A^*V - A^*WA^*$, where U, V, W are finite subsets of A^*. They also recognize subshifts of finite type, if we consider the bi-infinite words recognized (see [6]). They are heavily used to construct transducers and coding schemes adapted to constrained channels. When the output of the transducer is a local automaton, the decoding can be done with a sliding window, and the size of the window is bounded by the synchronization delay of the automaton. Finding adapted transducers with short synchronization delay in output, to get a short window in order to limit the error propagation, is one of the main goals when building codes for constrained channels (see for example [8],[6],[2]).

The local property means that all long enough blocks are synchronizing words, also called resolving blocks or reset sequences. A non-synchronizing sequence of a deterministic local automaton is a word which is the label of two paths ending in (and going through) different states. The synchronization delay is then equal to the length of one of the longest non-synchronizing sequence plus 1. For a given automaton $\mathcal{A}$ with N-states, this delay can be computed in a polynomial time by using the product automaton, whose states are pairs of states of $\mathcal{A}$. The product automaton restricted to pairs of distinct states of $\mathcal{A}$ has no cycle. The delay of a local deterministic automaton $\mathcal{A}$ is then the height of this directed acyclic graph plus 1. It is known that the delay is $O(N^2)$ for N-state local automata, and $O(N)$ for deterministic complete local automata. But it is was not known if this bound could be improved.

This problem can be related to a similar (but different) question about synchronizing sequences known as the Cerný-Pin conjecture. One has here to find automata with very long non-synchronizing sequences, (the locality implying that this length is bounded), where the Cerný problem is to show that there are short synchronizing words (of length at most $(N-1)^2$) for N-state complete synchronizing automata, (that is automata that admit at least one synchronizing word).

In two papers about locally testable languages ([4] and [5]) S. Kim, R. McNaughton and R. McCloskey conjectured that this synchronization delay is $O(N^{1.5})$ on a two-letter alphabet, and they gave an example of a family of automata leading to this bound. More precisely, their conjecture stated that if a locally testable automaton over a binary alphabet has N states then its order k (the smallest k for which the automaton is k-testable) is $O(N^{1.5})$. What we prove in this paper, that is relevant to this conjecture, is about a proper subset of the locally testable automata. We explicitly prove that the order of a local automaton over a binary alphabet is $\Omega(N^2)$, where N is the number of its states. This result disproves the conjecture by Kim et al. because the set of local automata is a subset of the set of locally testable automata. In order to prove our result, we give an example of a family of N-states local automata, that has a synchronization delay that is $\Omega(N^2)$. We also mention that the example, given in [2], of a family of N-state local automata on a two-letter alphabet with $\Omega(N^2)$ synchronization delay, is false.

It is easy to construct N-state automata with $\Omega(N^2)$ synchronization delay when the alphabet size is unbounded. This leads us to consider only the bounded case. We first give a general method to construct a local automaton on a two-letter alphabet from a local automaton on a r-letter alphabet, by encoding the r-letter alphabet in a circular code on the two-letter alphabet. We use this construction to prove, independently from the example we give after, that the complexity of the bound is the same in the case of a two-letter alphabet and in the case of a r-letter one, where $r \geq 2$. We then give in section 4 an example of automata which shows the main result, that is that the bound is $\Omega(N^2)$.

We thank the referee for helpful comments.

2 Background

We first make precise notations used to compare the complexities.

If f and g are two functions from $\mathbb{N}$ to $\mathbb{R}^+$, we say that $f \sim g$ if and only if $f = O(g)$ and $g = O(f)$. We say that $f = \Omega(g)$ if and only if there is a positive constant K such for all integer m, there is an integer $n > m$ such that $f(n) > Kg(n)$.

A finite automaton is said to be *local* if there are two integers k and d, with $0 \leq d \leq k$, such that if two finite paths of length k, $((p_i, a_i, p_{i+1}))_{0 \leq i \leq k-1}$ and $((p'_i, a_i, p'_{i+1}))_{0 \leq i \leq k-1}$, have the same label, then $p_d = p'_d$. A deterministic automaton is local if and only if the previous condition is satisfied with $d = k$. Deterministic finite automata have also been called definite automata in [9]. We call *synchronization delay* of a local automaton the smallest integer k satisfying the conditions of the definition.

A finite automaton is said to be *unambiguous* if for any states p and q (q may be equal to p), there are not two distinct equally labeled paths going from p to q.

The following properties are known (see for example [2] p.44-46 for proofs).

Proposition 1 *Let $\mathcal{A}$ be an automaton with a strongly connected graph. The two following properties are equivalent:*
1) the automaton $\mathcal{A}$ is local.
2) the automaton $\mathcal{A}$ does not admit two distinct equally labeled cycles.

In this proposition, 1) $\Longrightarrow$ 2) is true even if the graph is not strongly connected, and 1) $\Longleftrightarrow$ 2) remains true if one suppose that the automaton is unambiguous instead of being strongly connected. A local strongly connected automaton is unambiguous.

Proposition 2 *Let $\mathcal{A}$ be an N-state local automaton which is unambiguous or has a strongly connected graph. Its synchronization delay is upper bounded by $(N^2 - N)$.*

We briefly recall the proof given in [2] p.45.

Proof : We define $m = \frac{N(N-1)}{2}$ and $k = 2m$. We consider two paths of length k: $((p_i, a_i, p_{i+1}))_{0 \leq i \leq k-1}$ and $((p'_i, a_i, p'_{i+1}))_{0 \leq i \leq k-1}$, with the same label. There are at most m distinct pairs $\{p, q\}$ of distinct states. As $\mathcal{A}$ is local, it does not admit two equally labeled cycles. This implies that there are not two distinct indices i, j, with $0 \leq i, j \leq m$, such that $(p_i \neq p'_i, p_j \neq p'_j)$ and $(((p_i, p'_i) = (p_j, p'_j))\text{or}((p_i, p'_i) = (p'_j, p_j)))$. It follows that there is an index i, with $0 \leq i \leq m$, such that $p_i = p'_i$. There is also an index j with $m \leq j \leq k$ such that $p_j = p'_j$. As $\mathcal{A}$ is local and has a strongly connected graph, it is also unambiguous, and then $p_m = p'_m$. The condition of locality is satisfied with paths of length k and $d = m$. $\Box$

As a consequence, an N-state deterministic local automaton has a synchronization delay which is $O(N^2)$.

Proposition 3 *Let $\mathcal{A}$ be an N-state deterministic local automaton. Its synchronization delay is upper bounded by $\frac{N(N-1)}{2}$.*

Proof : We define $k = \frac{N(N-1)}{2}$. We consider two paths of length k and with the same label : $((p_i, a_i, p_{i+1}))_{0 \leq i \leq k-1}$ and $((p'_i, a_i, p'_{i+1}))_{0 \leq i \leq k-1}$. Like in previous proof, there is an index i, with $0 \leq i \leq k$, such that $p_i = p'_i$. As $\mathcal{A}$ is deterministic, we get $p_k = p'_k$. □

We recall that an automaton on an alphabet A is a *complete deterministic automaton* if each state admits exactly one outgoing edge labeled by each letter of the alphabet A. A proof of the following known result can be found in [2] p.46:

Proposition 4 *Let $\mathcal{A}$ be an N-state automaton which is complete deterministic and local. Its synchronization delay is upper bounded by $(N-1)$.*

3 Link with circular codes

We now present a general method to encode the letters of a local automaton on an alphabet of n letters into a code on a two-letter alphabet, in such a way that the composed automaton is still local. We will use circular codes.

Let A be an alphabet. A subset C of A^+ is said to be a *circular code* if for all $n, m \geq 1$ and $x_1, x_2, \ldots, x_n \in C$, $y_1, y_2, \ldots, y_m \in C$, and $p \in A^*$ and $s \in A^+$, the equalities

$$\begin{aligned} s x_2 \ldots x_n p &= y_1 y_2 \ldots y_m \\ x_1 &= ps \end{aligned}$$

imply

$$n = m, p = \epsilon, \text{ and } x_i = y_i \quad (1 \leq i \leq n)$$

Circular codes are codes such that words of A^* have at most one decomposition in the codewords on a cycle. Let $\mathcal{A}$ be an automaton on an alphabet A. We choose a state q of $\mathcal{A}$. The subset X_q of A^* of *first returns* to state q is defined as the set of labels of all paths going from state q to state q, without going through state q between the extremities. As the automaton is finite, the set X_q is rational.

We will call a 1-*pole automaton* an automaton which has the following property: there is a state q such that all cycles go through state q. The set of first returns to state q is then finite.

The link between local automata and circular codes is given in the two following known propositions (see for example [3] or [2] for a proof):

Proposition 5 *If $\mathcal{A}$ is local then X_q is a circular code, for any state q of $\mathcal{A}$. In the other direction, if C is a* finite *circular code, there is a* 1*-pole automaton $\mathcal{A}$ and a state q of $\mathcal{A}$ such that $X_q = C$. One can choose the flower automaton of the code C. If C is a finite circular code which is* *the set of first returns of a* 1*-pole unambiguous automaton, then this* 1*-pole automaton is local.*

A *very pure* monoid M is a monoid with the following property:

$$uv, vu \in M \Longrightarrow u, v \in M$$

Proposition 6 *A set $C \subset A^*$ is a circular code if and only if C^* is a very pure monoid and $CC \cap C = \emptyset$.*

An example of circular code is the subset $X_0 = \{AAC, AAT, ACC, ATC, ATT, CAG, CTC, CTG, GAA, GAC, GAG, GAT, GCC, GGC, GGT, GTA, GTC, GTT, TAC, TTC\}$ discovered by D. Arquès and C. Michel. It is the set of 20 trinucleotides having a preferential occurrence in the frame 0 of protein (coding) genes of both prokariotes and eukariotes. The reading frame is established by the ATG start trinucleotide (see [1]). The synchronization delay of its local flower automaton is 13, and it is the set of first returns of a 1-pole automaton on a 4-letter alphabet with 11 states. Similar circular codes also exist for the two other frames.

We now define the notion of composition of two codes and of composition of an automaton with a code. Let $Z \subset A^*$ and $Y \subset B^*$, where A and B are finite alphabets, be two codes together with a bijection β from B onto Z. Then β defines an injective morphism $\beta : B^* \longrightarrow A^*$. The set $X = \beta(Y)$ is obtained by *composition* of Y and Z and is denoted by $X = Y \circ Z$. The set X is obtained by coding the letters of Y by the corresponding words of Z. It is known that if Y and Z are composable circular codes, $X = Y \circ Z$ is a circular code (see [3]).

We will prove a similar result about a composition of an automaton with a circular code.

Let $\mathcal{A}$ be a an automaton on an alphabet B. Let Z be a finite circular code on an alphabet A with a bijection β from B onto Z. Then β defines an injective morphism $\beta : B^* \longrightarrow A^*$. We call *composition* of the automaton $\mathcal{A}$ with β, the automaton denoted by $\mathcal{A} \circ \beta$, and obtained by replacing each edge (p, b, q) of $\mathcal{A}$ by the edges:

$$p \xrightarrow{a_1} \bullet \xrightarrow{a_2} \bullet \xrightarrow{a_3} \bullet \cdots \xrightarrow{a_{n-1}} \bullet \xrightarrow{a_n} q$$

with $\beta(b) = a_1 a_2 \ldots a_n$, and where $(n-1)$ new intermediate states have been added.

If $\mathcal{A}$ is deterministic, we can define a deterministic version of the composition of $\mathcal{A}$ with β, when $Z = \beta(B)$ is a prefix circular code. Under this hypothesis, let p be state of $\mathcal{A}$ and let $(p, b_i, q_i)_{1 \leq i \leq s}$ be its outgoing edges. Let T_p be a labeled tree representing the prefix code $Z_p = \beta(\{b_1, \ldots, b_s\}) \subset Z$: the edges of T_p are labeled in the alphabet A, and each word of Z_p is the label of exactly one path from the root to a leaf. We label the root by p and each leaf corresponding to $\beta(b_i)$ by q_i. We now define $\mathcal{A} \circ_{det} \beta$ as the automaton obtained from $\mathcal{A}$ by replacing, for each state p, the outgoing edges

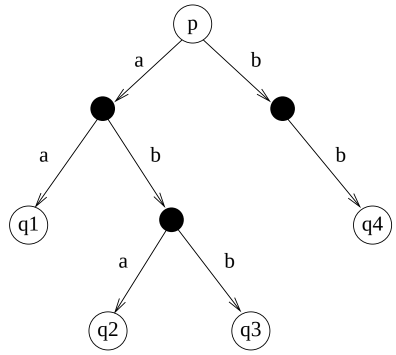

Figure 1: Tree T_p

of p in $\mathcal{A}$ by the edges of the tree T_p. The internal nodes of T_p will be new intermediate states in $\mathcal{A} \circ_{det} \beta$. Figure 1 shows the tree T_p when the outgoing edges of p are $(p, b_i, q_i)_{1 \leq i \leq 4}$, $A = \{a, b\}$, and $Z_p = (aa, aba, abb, bb)$.

Proposition 7 *If $\mathcal{A}$ is a local automaton that either has a strongly connected graph or is unambiguous, and if $Z = \beta(B)$ is a circular code, $\mathcal{A} \circ \beta$ is local. Moreover, if $\mathcal{A}$ is a local deterministic automaton and if Z is prefix circular, $\mathcal{A} \circ_{det} \beta$ is local and deterministic.*

Proof : We prove the first part of the proposition. The second one is a consequence of the first one, as $\mathcal{A} \circ_{det} \beta$ can be seen as a projection of $\mathcal{A} \circ \beta$.

We consider two distinct and equally labeled cycles of $\mathcal{A} \circ \beta$, one beginning at (and ending in) a state p, the other one beginning at (and ending in) a state q. We can suppose that $p \neq q$. Without loss of generality we also can assume that at least one of them is also a state of $\mathcal{A}$ (not an intermediate added state). If they are both states of $\mathcal{A}$, we get in $\mathcal{A}$ two distinct equally labeled cycles, which contradicts the locality of $\mathcal{A}$. We now suppose that p is a state of $\mathcal{A}$ and q is not. We can remark that each cycle of $\mathcal{A} \circ \beta$ goes through a state of $\mathcal{A}$. Let r be the first state belonging to the states of $\mathcal{A}$ in the cycle beginning at q. This cycle is composed of a path labeled u from q to r, concatenated to a path labeled v from r to q. The word uv is also the label of the other cycle beginning at p. We get $uv, vu \in Z^*$. As Z is a circular code, Z^* is a very pure monoid. We get $u, v \in Z^*$. This forces p, q, r to be states of $\mathcal{A}$, which concludes the proof. $\square$

We will use an encoding of the alphabet of $\mathcal{A}$ in a particular class of circular codes: the comma-free codes. A subset C of A^+ is a *comma-free* code if and only if, for all $w \in C^+$, $u, v \in A^*$,

$$uwv \in C^* \Longrightarrow u, v \in C^*$$

We refer to [3] for this notion and the following result due to Golomb et al. and Eastman:

Theorem 1 *For any alphabet A with r letters for any odd integer $m \geq 1$, there exits a comma-free code $C \subset A^m$ such that $Card(C) = l_m(r)$, where $l_m(r)$ is number of conjugacy classes of primitive words of length m in A^*.*

Moreover, we have

$$l_m(2) \sim \frac{2^m}{m}.$$

This theorem implies that it is possible to find a comma-free (thus circular) code on a two-letter alphabet composed of l words of length m with $m \sim \log(l)$. As the codewords have the same length, the code is also a prefix code. By considering the case of an alphabet A of size r, we get the following result:

Proposition 8 *Let r be an fixed integer and let f be an increasing function from $\mathbb{N}$ to $\mathbb{R}^+$ such that for any positive constant c, $f(N) \sim f(cN)$. If the synchronization delay of an N-state local deterministic automaton on a r-letter alphabet is $\Omega(f(N))$, then the synchronization delay of N-state local deterministic automaton on a 2-letter alphabet is $\Omega(f(N))$.*

Proof : Let $\mathcal{A}_n$ be a family of N-state local deterministic automata on a r-letter alphabet, with $N \sim n$, and with a synchronization delay at least $k(N)$, where $k(N)$ is $\Omega(f(N))$. Let $\mathcal{B}_n$ the family $\mathcal{A}_n \circ_{det} \beta_n$ of deterministic automata on the two-letter alphabet, where β_n defines a coding from the alphabet of $\mathcal{A}_n$ in a prefix circular code of words of a fixed length m on the two-letter alphabet. As $\mathcal{A}_n$ is deterministic, the number of its edges is $F \leq rN$. The number of states of $\mathcal{B}_n$ is

$$N' \sim (N + F)m \sim N \times r \times m.$$

The synchronization delay of $\mathcal{B}_n$ is greater than $(k(N) - 1)m$, at least

$$(k(\frac{N'}{rm}) - 1)m.$$

As r and m are constants, the synchronization delay of $\mathcal{B}_n$ is then $\Omega(f(\frac{N'}{rm}))$ $= \Omega(f(N'))$. $\square$

This proposition applies in particular in the case where $f(N) = N^2$. This proposition shows that the complexity of the synchronization delay in the case of a 2-letter alphabet is the same as in the case of a r-letter one, for any $r \geq 2$. In the next section we give an example of a family of automata with $\Omega(N^2)$ synchronization delay on a two-letter alphabet.

4 Upper bound of the synchronization delay

We now go to the case of a fixed alphabet with two letters $A = \{a, b\}$. Let $\mathcal{A}_n$ be the following family of N-state local deterministic automata on the

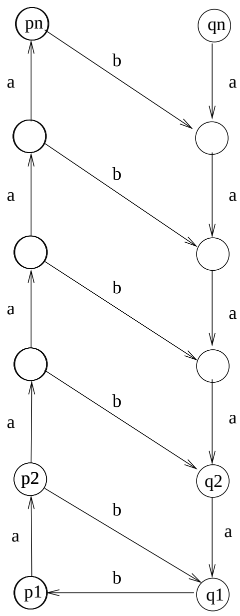

Figure 2: Ladder automaton

alphabet A. The set of states of $\mathcal{A}_n$ is $\{p_1, p_2, \ldots, p_n, q_1, q_2, \ldots, q_n\}$. We then have $N = 2n$. The edges of $\mathcal{A}_n$ are

$$\begin{array}{lcl} q_k & \xrightarrow{a} & q_{k-1}, \quad 2 \leq k \leq n \\ p_k & \xrightarrow{a} & p_{k+1}, \quad 1 \leq k \leq n-1 \\ p_k & \xrightarrow{b} & q_{k-1}, \quad 2 \leq k \leq n \\ q_1 & \xrightarrow{b} & p_1 \end{array}$$

The automaton $\mathcal{A}_n$ is given in Figure 2.

For a deterministic automaton, we call a *non-synchronizing* sequence a finite word u such that there are two paths labeled by u, $(p_i, a_i, p_{i+1})_{0 \leq i \leq k-1}$ and $(p'_i, a_i, p'_{i+1})_{0 \leq i \leq k-1}$, with $p_i \neq p'_i$ for all indices i.

Proposition 9 *The above defined N-state automaton $\mathcal{A}_n$ is local and its synchronization delay is $\Omega(N^2)$.*

Proof : We first prove that $\mathcal{A}_n$ is local. We can prove this directly by using the definition of local automata. We give here a proof that uses circular codes. We remark that $\mathcal{A}_n$ is a 1-pole automaton and we can choose state p_1 as pole. Then $\mathcal{A}_n$ is local if and only if the finite code of first returns to state p_1 is circular. This code is $C = \{a^k b a^{(k-1)} b, 1 \leq k \leq n\}$. Let us suppose that this code is not circular. We then have two circular decompositions:

$$\begin{array}{rcl} s x_2 \ldots x_r p & = & y_1 y_2 \ldots y_m \\ x_1 & = & ps \end{array}$$

with $x_1, x_2, \ldots, x_r \in C$, $y_1, y_2, \ldots, y_m \in C$ and $p \in A^+$ and $s \in A^+$. These two decompositions in words $a^k ba^{(k-1)}b$ of C are :

$$\cdots \underbrace{a^k ba^{(k-1)}b}_{y_1}\underbrace{a^{(k-2)}ba^{(k-3)}b}_{y_2}\cdots\underbrace{a^{(k-2m+2)}ba^{(k-2m+1)}b}_{y_m}\cdots$$

$$\overbrace{\cdots a^k b}^{ps}\overbrace{a^{(k-1)}ba^{(k-2)}b}^{x_2}a^{(k-3)}b\overbrace{\cdots a^{(k-2m+2)}b}^{x_r}\overbrace{a^{(k-2m+1)}b\cdots}^{ps}$$

and we get $r = m$. This implies that $s = a^k b$ and $p = a^{(k-2m+1)}b$ and ps cannot belong to C.

By considering the paths beginning at states $p1$ and q_n labeled by $u = a^{(n-1)}ba^{(n-2)}b \cdots ab$, one sees that u is a non-synchronizing sequence. The synchronization delay of $\mathcal{A}_n$ is then greater than or equal to the length of u plus 1, that is to $\frac{n(n+1)}{2}$. As $N = 2n$, it is $\Omega(N^2)$. $\square$

We now briefly describe the relationship between this example and the conjecture by Kim et al. (see also the first section). Let us consider the above automaton with p_1 as initial state and the sole accepting state. We can discard the state q_n so that the automaton is strongly connected. The language of finite words recognized by this automaton is locally testable but not k-testable with $k = ((n^2 - n)/2)$. So the example disproves the conjecture by Kim et al.

References

[1] D. Arquès and C. Michel. A complementary circular code in the protein coding genes. *J. Theor. Biol.*, 182:45–58, 1996.

[2] M.-P. Béal. *Codage Symbolique.* Masson, 1993.

[3] J. Berstel and D. Perrin. *Theory of Codes.* Academic Press, 1985. (also available on http://www-litp.ibp.fr/berstel/LivreCodes).

[4] S. Kim and R. McNaughton. Computing the order of a locally testable automaton. *SIAM J. Comput.*, 23(6):1193–1215, 1994.

[5] S. Kim, R. McNaughton, and R. McCloskey. A polynomial time algorithm for the local testability problem of determinitic finite automata. *I.E.E.E. Trans. Comput.*, 40:1087–1093, 1991. (see also L.N.C.S. 382 (1989) pp. 420-436).

[6] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding.* Cambridge, 1995.

[7] A. D. Luca and A. Restivo. A characterization of strictly locally testable languages and its application to subsemigroups of a free semigroup. *Inform. and Control*, 44:300–319, 1980.

[8] B. Marcus, P. Siegel, and J. Wolf. Finite-state modulation codes for date storage. *IEEE J. Selected Areas Commun.*, 10(1):5–37, 1992.

[9] M. Perles, M. O. Rabin, and E. Shamir. The theory of definite automata. *I.E.E.E. Trans. Electr. Comp.*, EC 12:233–243, 1963.