# ASYNCHRONOUS CELLULAR AUTOMATA FOR INFINITE TRACES*

**Paul GASTIN**
Université PARIS 6
LITP, Institut Blaise Pascal
4, place Jussieu
75 252 PARIS CEDEX 05
FRANCE
gastin@litp.ibp.fr

**Antoine PETIT**
Université PARIS SUD
LRI, URA CNRS 410
Bât. 490
91 405 ORSAY CEDEX
FRANCE
petit@lri.lri.fr

**Abstract:** The notion of infinite traces has been proposed to generalize both the finite traces defined by Mazurkiewicz and the infinite words. The family of recognizable languages of infinite traces has been introduced by means of recognizing morphisms and characterized by c-rational expressions. In this paper, we provide another characterization of this family using asynchronous cellular automata (which carry the most intuitive idea of finite state concurrent machines). To this purpose, we give effective constructions for c-rational operations on these automata, which are of independent interest.

## 1. Introduction

The concept of traces has been introduced by Mazurkiewicz [Maz77] as a suitable semantics for non-sequential processes. Let us refer, for instance, surveys [Maz86, AR88, Per89] or the monograph [Die90] where also extensive bibliographies on the subject are given.
In order to describe the behaviours of non sequential processes which never terminate, the monoid $\mathbb{C}$ of complex traces has been proposed [Die91]. In this monoid, the subset $\mathbb{R}$ of real traces is of first importance since it turns out that a complex trace is simply a real trace together with an alphabetic information. In fact, previously to the definition of $\mathbb{C}$, real traces were already introduced and intensively studied [BMP89, Gas90, Kwi90, GR91].
Recognizable languages describe the behaviours of finite state systems and hence form one of the basic families of a monoid. As regards infinite traces, the family of recognizable languages has been defined by recognizing morphisms in [Gas91] and characterized by c-rational expressions in [GPZ91] and [DGP91]. But, until now, it suffers from lack of characterizations by means of finite automata. The aim of this paper is to remedy this deficiency. For lack of space, we focus on automata on recognizable real trace languages only.
For recognizable languages of finite traces, Zielonka introduced *asynchronous* and *asynchronous cellular* automata which are major special classes of trace automata [Zie87, Zie89, CMZ89]. These automata are more suited for the description of concurrent processes since, thanks to their decentralized control, they allow concurrent executions of independent actions.
A Büchi automaton for infinite words is simply a classical finite word automaton together with acceptance conditions for infinite paths [Eil74, Tho90, PP91]. Hence, we will extend

* This work has been partly supported by the ESPRIT Basic Research Actions N° 3166 (ASMICS) and N° 3148 (DEMON) and by the PRC $C^3$ and Math-Info.

asynchronous cellular automata to accept infinite real traces using a suitable Büchi like acceptance condition for infinite paths. In order to obtain a class of automata which are suitable for infinite traces, we will use a decentralized acceptance condition. We show that these Büchi asynchronous cellular automata accept recognizable real trace languages.
Conversely, we prove that each recognizable real trace language is accepted by some Büchi asynchronous cellular automaton. To this purpose, we use effective constructions to show that the class of Büchi asynchronous cellular automata is closed under union, intersection, concatenation and a restricted infinite iteration.
Therefore, we extend the deep Zielonka's theorem to infinite traces with a characterization of the family of recognizable real trace languages by means of Büchi asynchronous cellular automata.

## 2. Preliminaries

In this section, we recall definitions and results on traces which are relevant in this paper. For more detailed discussions, the reader is referred to [Maz86, AR88, Per89, Die90] as concerns finite traces, to [Gas91, GPZ91] as concerns real traces.

### 2.1. Dependence graphs and traces

In the following (X,D) denotes a finite dependent alphabet, i.e. X is a finite alphabet together with a reflexive and symmetric dependence relation $D \subseteq X \times X$. We denote by $I = X \times X \setminus D$ the independence relation which is the complement of D.
A dependence graph over (X,D) is (an isomorphism class of) a labelled acyclic graph $[V,E,\lambda]$ where V is the (at most) countable set of vertices, $E \subseteq V \times V$ is the set of arcs, $\lambda: V \to X$ is the labelling and which satisfies:

i) the induced partial order $(V,E^*)$ is well-founded: there does not exist an infinite descending chain i.e. an infinite sequence $(x_i)$ such that $(x_{i+1}, x_i) \in E$ for all i.
ii) edges are between dependent vertices:
$$\forall\ x,y \in V, \qquad ((\lambda(x),\lambda(y)) \in D \Leftrightarrow x = y \text{ or } (x,y) \in E \text{ or } (y,x) \in E$$

Then, for any a in X, the restriction to the set of vertices with label a is well-ordered. This allows to think of dependence graphs by standard representations where the vertices are pairs (a,i) with $a \in X$ and i is a countable ordinal. The set of dependence graphs is denoted by $\mathbb{G}(X,D)$ or simply by $\mathbb{G}$ when there is no confusion. It is a monoid for the operation $[V_1,E_1,\lambda_1][V_2,E_2,\lambda_2] = [V,E,\lambda]$ where $[V,E,\lambda]$ is the disjoint union of $[V_1,E_1,\lambda_1]$ and $[V_2,E_2,\lambda_2]$ together with new arcs $(p_1,p_2)$ for all vertices $p_1 \in V_1$, $p_2 \in V_2$ such that $(\lambda_1(p_1),\lambda_2(p_2)) \in D$. The neutral element is the empty graph $1 = [\emptyset,\emptyset,\emptyset]$.
The concatenation is generalized to an infinite product as follows. Let $(g_i)$ be any sequence of dependence graphs. Then $g = g_1 g_2 \ldots g_i \ldots \in \mathbb{G}$ is defined as the disjoint union of the $g_i$ with the obvious new arcs from $g_i$ to $g_j$ for $i<j$ whenever the vertices have dependent labels. Note that concatenation and infinite product are associative operations.
Using these operations, we relate words and dependence graphs. We define the mapping $\varphi: X^\infty \to \mathbb{G}$ by $\varphi(a) = [\{x\},\emptyset,x \mapsto a]$ for all a in X and $\varphi(a_1a_2\ldots) = \varphi(a_1)\varphi(a_2)\ldots$ for all finite or infinite word $a_1a_2\ldots$ in $X^\infty$. Note that $\varphi$ is not injective, for instance $\varphi(ab) = \varphi(ba)$ iff $(a,b) \in I$. We define an equivalence $\sim_I$ (or simply $\sim$ when there is no confusion) on $X^\infty$ by

$u \sim v$ if $\varphi(u) = \varphi(v)$ for all $u,v$ in $X^\infty$. More precisely, $\sim$ restricted to finite words is the smallest congruence such that $ab$ and $ba$ are equivalent when $(a,b) \in I$. Whereas two infinite words are equivalent if and only if for every finite prefix $f$ of one of them, there exists a finite word $g$ such that $fg$ is equivalent to a (finite) prefix of the other. It turns out that $\varphi$ is a surjective morphism from the free monoid $X^*$ onto the monoid $\mathbb{IF}\mathfrak{G}(X,D)$ of finite dependence graphs. The monoid of finite traces introduced by A. Mazurkiewicz is precisely $X^*/\sim = \mathbb{IF}\mathfrak{G}(X,D)$ which is denoted by $\mathbb{IM}(X,D)$ or simply by $\mathbb{IM}$ throughout.

### 2.2. Infinite traces

In a similar way, we will define the monoid of infinite traces from the monoid $\mathfrak{G}$. But it turns out that $\varphi : X^\infty \to \mathfrak{G}$ is neither surjective nor a morphism. The set $\varphi(X^\infty) = X^\infty/\sim$ of dependence graphs which can be represented by words is denoted by $\mathbb{IR}(X,D)$ or simply by $\mathbb{IR}$ when there is no confusion. Its elements are called real dependence graphs or real traces. $\mathbb{IR}$ is characterized as follows: a dependence graph $[V,E,\lambda]$ is a real one iff the past $p{\downarrow} = \{q \in V / (q,p) \in E^*\}$ is finite for every vertex $p$ of $V$. Unfortunately $\mathbb{IR}$ is not a submonoid of $\mathfrak{G}$. For instance if $a$ and $b$ are two dependent letters, the concatenation of the two real graphs $a^\omega$ and $b^\omega$ is not a real graph. More precisely, for a real dependence graph $g = [V,E,\lambda]$, let $\mathrm{alph}(g) = \lambda(V)$ be the set of letters which occur in $g$ and let $\mathrm{alphinf}(g) = \{a \in X / |\lambda^{-1}(a)| = \omega\}$ be the set of letters which occur infinitely often in $g$. Then the concatenation of two real graphs $r$ and $s$ is a real graph if and only if $\mathrm{alphinf}(r) \times \mathrm{alph}(s) \subseteq I$. In particular the concatenation of a finite trace and a real trace is always a real trace. Besides, an infinite product of finite traces is always a real trace. Note that in this paper we **never** use a concatenation or an infinite product of real traces which yield a **non** real trace.

Let $\eta : \mathbb{IM} \to S$ be a morphism into a finite monoid. A real trace language $R \subseteq \mathbb{IR}$ is recognized by $\eta$ if for any (finite or infinite) sequence $(r_i) \subseteq \mathbb{IM}$, we have

$$r_1 \cdot r_2 \cdot r_3 \ldots \in R \Rightarrow \eta^{-1}\eta(r_1) \cdot \eta^{-1}\eta(r_2) \cdot \eta^{-1}\eta(r_3) \ldots \subseteq R$$

We denote by $\mathrm{Rec}\,\mathbb{IR}$ (resp. $\mathrm{Rec}\,\mathbb{IM}$) the family of languages of real traces (resp. finite traces) which are recognized by some morphism. Note that if the independence relation $I$ is empty, these definitions reduce to the classical ones for recognizable word languages. In this case, the families are simply denoted by $\mathrm{Rec}\,X^\infty$ (resp. $\mathrm{Rec}\,X^*$).

In this paper, we use two important results on recognizable real trace languages [Gas91]. The first one relates traces and words: a real trace language $R \subseteq \mathbb{IR}$ is recognizable if and only if $\varphi^{-1}(R)$ is a recognizable word language. The second one gives a rational expression of a real trace language $R \subseteq \mathbb{IR}$ recognized by a morphism $\eta : \mathbb{IM} \to S$. For $s \in S$, let $R_s = n^{-1}(s)$. Then

$$R = \bigcup_{(s,e) \in P} R_s R_e^\omega \text{ where } P = \{(s,e) \in S^2 / se = s,\ e^2 = e \text{ and } R_s R_e^\omega \subseteq R\}.$$

### 2.3. Automata on finite traces

In a general monoid $M$, a (non-deterministic) $M$-automaton [Ber79] is a tuple $\mathcal{A} = \langle Q,\delta,S,F\rangle$ where $Q$ is a finite set of states, $S$, $F \subseteq Q$ are the sets of initial (starting) and final states, $\delta : Q \times M \to \mathcal{P}(Q)$, $\delta(q,m) = qm$ is the transition function which verifies $q \in q1$ and $q(mn) = (qm)n$ for all $q \in Q$ and $m,n \in M$. The recognized set is $L(\mathcal{A}) = \{m \in M / Sm \cap F \neq \emptyset\}$. For finite traces, if such an automaton $\mathcal{A}$ recognizes a trace language $T$, $\mathcal{A}$ can also be considered as a word automaton which recognizes $\varphi^{-1}(T)$. Conversely a word automaton can be viewed as

a trace automaton provided the I-diamond criterion is verified: $qab = qba$ for any $q \in Q$ and $(a,b) \in I$. In fact it suffices in this case to define $qt = qx$ for any trace t and any word x in $\varphi^{-1}(t)$ to obtain a trace automaton (see e.g. [Die90]).
Two major subclasses of trace automata, namely asynchronous and asynchronous cellular automata, were introduced by Zielonka [Zie87,Zie89]. These automata have a distributed control structure and they can thus perform actions concurrently. Zielonka's theorem states that a trace language $T \subseteq \mathbb{M}$ is recognizable if and only if it is recognized by some deterministic asynchronous (cellular) automaton. In this paper we focus on asynchronous cellular automata but our results can be easily extended to asynchronous automata. The definition of asynchronous cellular automata will be obtained as a special case of the definition of Büchi asynchronous cellular automata given in next section.

## 3. Büchi asynchronous cellular automata

Although the set of finite and infinite words forms a monoid, the right notion of automata in this monoid is not the notion of M-automata presented in Section 2.3. Recall that an automaton on infinite words is a classical finite $X^*$-automaton together with acceptance conditions for infinite paths [Eil74,Tho90,PP91]. These conditions are on the set $f(P)$ of states which are repeated infinitely often in the path P. More precisely, in a Büchi automaton, a set R of repeated states is specified and a path P is Büchi-accepted if $f(P)$ contains some element of R. Büchi's theorem states that a word language $L \subseteq X^\infty$ is recognizable if and only if it is recognized by some Büchi automaton. It is well known that in this result, the Büchi automaton cannot be chosen deterministic (in other words the class of word languages recognized by deterministic Büchi automata forms a proper subclass of the family of recognizable word languages).

The principal aim of this paper is to extend Büchi's and Zielonka's theorems to languages of infinite traces and to (non-deterministic) Büchi asynchronous cellular automata. To this purpose we need first a good notion of Büchi asynchronous cellular automata.

Let us first introduce some notations in order to lighten formulas. Let $(Q_z)_{z\in Z}$ be a family of sets. For $Y \subseteq Z$, we denote by $Q_Y$ the cartesian product $\prod_{y\in Y} Q_y$ and if $q = (q_z)_{z\in Z} \in Q_Z$, we denote by $q_Y$ the subfamily $(q_y)_{y\in Y} \in Q_Y$. Finally, $\mathrm{Cell}(Q_X) = \{ \prod_{z\in Z} R_z \,/\, R_z \subseteq Q_z \text{ for all } z \in Z\}$ is the set of cells of $Q_Z$.

A **Büchi asynchronous cellular automaton** (BACA for short) is a tuple $\mathcal{A} = \langle (Q_a)_{a\in X}, (\delta_a)_{a\in X}, S, F, \mathcal{R}\rangle$ where, for each $a \in X$, $Q_a$ is the finite set of local states for a, $\delta_a$ from $Q_{D(a)}$ to $\mathcal{P}(Q_a)$ is the local transition function for a, $S,F \subseteq Q_X$ are the sets of starting and final states respectively and $\mathcal{R} \subseteq \mathrm{Cell}(Q_X)$ is the acceptance table.
The best way to describe how an asynchronous automaton works in a distributed way is probably to define paths for dependence graphs. See for instance [Tho89, Tho91] where this approach has been used for asynchronous automata on finite traces. For lack of space, we propose here a more classical (equivalent) formulation using global paths.
The global transition function $\Delta : Q_X \times X \to \mathcal{P}(Q_X)$ of $\mathcal{A}$ is defined by:

$$q' \in \Delta(q,a) \text{ if } q'_a \in \delta_a(q_{D(a)}) \text{ and } q'_{X\setminus\{a\}} = q_{X\setminus\{a\}}$$

Note that, with its global transition function, a BACA can be viewed as a usual word automaton which turns out to be I-diamond.

A finite or infinite path P is a sequence $q_0,a_1,q_1,a_2,q_2,\ldots$ such that $q_i \in \Delta(q_{i-1},a_i)$ for all $i \geq 1$. The label of P is the word $\ell(P) = a_1a_2\ldots$, the starting state of P is $s(P) = q_0$. For any $a \in X$, we define the final local information $f_a(P)$ by:

$f_a(P) = \{q_{i,a} / i = \inf \{j \geq 0 / \forall\ k > j,\ a_k \neq a\}\}$ is the singleton which consists of the final local state for a, if P has finitely many transitions labelled by a

and

$f_a(P) = \{p_a \in Q_a / |\{i / a_i = a \text{ and } q_{i,a} = p_a\}| = \omega\}$ is the set of local states for a repeated infinitely often, otherwise.

From the definition of a BACA, it is easy to verify that, in both cases, for each $a \in X$, $f_a(P) = \{p_a \in Q_a / |\{i / q_{i,a} = p_a\}| = \omega\}$. We denote by $f(P) = \prod_{a\in X} f_a(P)$ the final information of P.

Note that, if $P = q_0,a_1,q_1,\ldots,a_n,q_n$ is a finite path, then $f(P) = \{q_n\}$.

Finally, a path P is accepted if $s(P) \in S$ and either P is finite and $f(P) \cap F \neq \emptyset$ or P is infinite and $f(P) \supseteq R$ for some $R \in \mathcal{R}$. The word and trace languages accepted by $\mathcal{A}$ are respectively

$L(\mathcal{A}) = \{u \in X^\infty / u = \ell(P) \text{ for some accepted path } P\}$ and

$T(\mathcal{A}) = \{t \in \mathbb{R} / t = \varphi(\ell(P)) \text{ for some accepted path } P\} = \varphi(L(\mathcal{A}))$.

The notion of asynchronous cellular automata (ACA for short) introduced by Zielonka is simply the notion of BACA with empty acceptance tables ($\mathcal{R} = \emptyset$).

The remainder of this section will explain and justify this definition of BACA.

Since the global transition function $\Delta$ of a BACA is I-diamond, we have for each $q \in Q_X$, $\Delta(q,u) = \Delta(q,v)$ if u and v are two equivalent finite words. This fact is strengthened and extended to finite and infinite words in the following proposition.

**Proposition 3.1.** Let u,v be two equivalent finite or infinite words and let $\mathcal{A}$ be a BACA. Then, for each path P such that $\ell(P) = u$, there exists a path P' such that $\ell(P') = v$, $s(P') = s(P)$ and $f(P') = f(P)$.

**Proof.** Let us first introduce a notation. For a path $P = q_0,a_1,q_1,a_2,q_2,\ldots$ and a letter $a \in X$, $State_a(P) = \{q_{i,a} / i \geq 0\}$ is the set of local states and $State(P) = \prod_{a\in X} State_a(P)$.

The proof of this proposition is based on the two following claims. The first one lies on the distributed structure of an asynchronous cellular automaton and is well-known.

Claim 1. Let x,y be two equivalent finite words of $X^*$, then for any path P in $\mathcal{A}$ such that $\ell(P)=x$, there exists a path P' in $\mathcal{A}$ such that $\ell(P')=y$ and verifying $s(P)=s(P')$, $f(P)=f(P')$ and $State(P)=State(P')$.

The second claim relates two equivalent infinite words. It can be proved using the characterization of equivalent infinite words by finite prefixes recalled in Section 2.1.

Claim 2 [DGP91]. Let u,v be two equivalent infinite words, then there exist four sequences of finite words $(u_i)$, $(v_i)$, $(x_i)$ and $(y_i)$ such that $u=u_1u_2u_3\ldots$, $v=v_1v_2v_3\ldots$ and for any $i \geq 1$, $u_i \sim x_iy_i$ and $v_i \sim y_{i-1}x_i$ (with $y_0=1$).

Let u,v be two equivalent infinite words, let q be a state of $\mathcal{A}$ and let P be a path such that $\ell(P) = u$ and $s(P) = q$. Let $(u_i)$, $(v_i)$, $(x_i)$ and $(y_i)$ be the sequences of finite words defined by

Claim 2 and let $(P_i)$ be the sequence of finite paths such that $P=P_1P_2P_3\ldots$ and $\ell(P_i)=u_i$ for $i\geq 1$. From Claim 1 applied to the words $u_i$ and $x_iy_i$ and to the path $P_i$, there exists a path $R_iS_i$ such that $\ell(R_i)=x_i$, $\ell(S_i)=y_i$, $s(P_i)=s(R_iS_i)$, $f(P_i)=f(R_iS_i)$ and $State(P_i)=State(R_iS_i)$. Applying once more Claim 1 to the words $y_{i-1}x_i$ and $v_i$ and to the path $S_{i-1}R_i$ (where $S_0$ is the empty path $S_0=q$), there exists a path $P'_i$ such that $\ell(P'_i)=v_i$, $s(P'_i)=s(S_{i-1}R_i)$, $f(P'_i)=f(S_{i-1}R_i)$ and $State(P'_i)=State(S_{i-1}R_i)$ (see Figure 3.1).

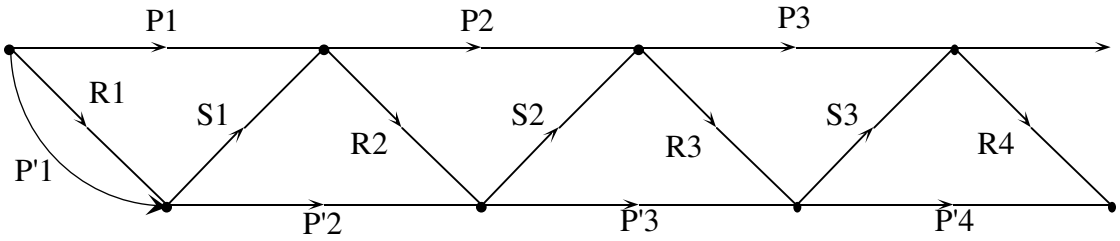


Figure 3.1

Now it is obvious that $P'=P'_1P'_2P'_3\ldots$ is an infinite path in $\mathcal{A}$ such that $\ell(P')=v$. For any letter $a \in X$, it holds $q_a \in f_a(P)$ iff there is an infinite number of paths $P_i$ s.t. $q_a \in State_a(P_i)$. Hence from the equalities above between the sets State(-) we obtain $q_a \in f_a(P)$ iff there is an infinite number of paths $R_iS_i$ s.t. $q_a \in State_a(R_i) \cup State_a(S_i)$ iff there is an infinite number of $P'_i$ s.t. $q_a \in State_a(P'_i)$ iff $q_a \in f_a(P')$. Then, for any letter $a \in X$, $f_a(P) = f_a(P')$ and the proposition is proved.
Note that the paths P and P' have very tight links (see Figure 3.1) which can be formalized with the notion of equivalent paths introduced in [GP91]. ◊

The following trivial corollary explains why BACA are trace automata.

**Corollary 3.2.** Let $\mathcal{A}$ be a BACA. Then, $L(\mathcal{A})$ is closed, i.e. $L(\mathcal{A}) = \varphi^{-1}(T(\mathcal{A}))$.

Informally, an infinite path P of a BACA is recognized if there exists an element $R = \prod_{a\in X} R_a$ in the acceptance table such that, for all $a \in X$ and $r_a \in R_a$, infinitely many states of P have $r_a$ as local state. The next remark claims that it would not have been possible to specify only repeated local states for some letters or even to specify one repeated local state for each letter.

**Remark 3.3.** Let $X=\{a,b\}$ with $(a,b)\in I$. If a BACA $\mathcal{A}$ recognizes the trace language $\{(ab)^\omega\}$ then each useful table $(R_a,R_b)$ in the acceptance table of $\mathcal{A}$ verifies $|R_a| \geq 2$ and $|R_b| \geq 2$. Indeed, let $\mathcal{A}$ be a BACA recognizing $\{(ab)^\omega\}$ and suppose that there exists a "useful" table $R = R_a \times R_b$ in the acceptance table $\mathcal{R}$ with, for instance, $|R_a| \leq 1$. Since R is "useful" there exists an infinite path P such that $R \subseteq f(P)$. If $|R_a| = 0$ then the path P' obtained from P by removing transitions with label a is also accepted. If $R_a = \{r_a\}$, we can write $P = P'P''$ with $f_a(P') = r_a$. Then the path $P'P'''$ is also accepted where $P'''$ is obtained from $P''$ by removing transitions with label a. In both cases, we obtain an accepted path with a label in $(a+b)^*b^\omega$ which is in contradiction with the fact that $\mathcal{A}$ recognizes the trace language $\{(ab)^\omega\}$.

The following proposition asserts that the (deterministic) BACA, considered as word automata, have the same expressive power as the classical (deterministic) Büchi automata. This explains why we legitimaly called them **Büchi** automata.

**Proposition 3.4.** Let $\mathcal{A}$ be a (deterministic) BACA, then there exists a classical (deterministic) Büchi automaton $\mathcal{A}'$ recognizing the word language $L(\mathcal{A})$.

**Proof.** It will be useful to introduce the technical notion of generalized Büchi automata on words. Such an automaton is a 5-tuple $\langle Q, \eta, S, F, \mathcal{R}\rangle$ where $\langle Q, \eta, S, F\rangle$ is a classical finite automaton and $\mathcal{R} \subseteq \mathcal{P}(Q)$ is a table of state sets. An infinite path P is accepted by such an automaton if the set $f(P)$ of states which occur infinitely often in P **contains** some member of $\mathcal{R}$. Note that a Büchi automaton is nothing else but a generalized Büchi automaton where $\mathcal{R}$ reduces to singletons. Nevertheless the expressive power for deterministic as well as non deterministic automata on word languages is not changed. To prove this result, we may reduce ourselves to generalized Büchi automata where $\mathcal{R}$ is constituted of one element since the classes of deterministic Büchi automata and non deterministic Büchi automata are both closed by union. More precisely the proof follows directly from the next easy claim (left to the reader).

> Claim. Let $\langle Q, \eta, S, F, \{R\}\rangle$ be a (deterministic) generalized Büchi automaton. Then the (deterministic) Büchi automaton $\langle Q\times\mathcal{P}(R), \eta', S', F', R'\rangle$ recognizes the same word language when $S' = S\times\{\emptyset\}$, $F' = F\times\mathcal{P}(R)$, $R' = Q\times\{R\}$ and $(q',K') \in \eta'((q,K),a)$ if $q' \in \eta(q, a)$ and $K' = \begin{cases}(K \cup \{a\}) \cap R & \text{if } K \neq R\\ \{a\} \cap R & \text{if } K = R\end{cases}$.

Now, let $\mathcal{A}=\langle (Q_a)_{a\in X}, (\delta_a)_{a\in X}, S, F, \mathcal{R}\rangle$ be a BACA. Let $\mathcal{A}'$ be the generalized Büchi automaton on words $\langle Q_X, \Delta, S, F, \mathcal{R}'\rangle$, with

$$\mathcal{R}' = \{R' \in \mathcal{P}(Q_X) \,/\, \exists\, R \in \mathcal{R} \text{ s.t. } \forall\, a \in X, \forall\, q_a \in R_a, \exists\, r' \in R' \; r'_a = q_a\}.$$

We claim that $L(\mathcal{A}') = L(\mathcal{A})$.
Since the second component of $\mathcal{A}'$ is useless for finite words, it is clear that $\mathcal{A}$ and $\mathcal{A}'$ recognize the same finite words. Hence, it remains to study the case of infinite words.
Let $P = q_0,a_1,q_1,a_2,q_2,\ldots$ be an infinite path accepted by the BACA $\mathcal{A}$. We prove that the set R' of states which occur infinitely often in P is in the table $\mathcal{R}'$. Suppose that P is accepted by some table $R \in \mathcal{R}$ i.e. $R \subseteq f(P)$. Let $a \in X$ and let $q_a \in R_a$, since $q_a$ is repeated infinitely often in P there exists a state q' such that, $q'_a = q_a$, and $q' = q_i$ for infinitely many integers i. Hence $q' \in R'$.
Conversely suppose that $P = q_0,a_1,q_1,a_2,q_2,\ldots$ is accepted by an element R' of $\mathcal{R}'$. This element R' has been constructed from an element R of $\mathcal{R}$, we will show that R accepts P in $\mathcal{A}$. By definition we get: $\forall\, r' \in R'$, $\left|\{i \,/\, q_i = r'\}\right| = \omega$ (1). Now let $a \in X$ and let $r_a \in R_a$, by definition of $\mathcal{R}'$, there exist an element $r' \in R'$ such that $r'_a = r_a$. If we apply (1) to this particular r', we get $\left|\{i \,/\, q_i = r'\}\right| = \omega$ whence $\left|\{i \,/\, q_{i,a} = r_a\}\right| = \omega$ and therefore the path P is accepted by R. ◊

From Büchi's theorem and Proposition 3.4, we deduce that, if $\mathcal{A}$ is a BACA, the language $L(\mathcal{A})$ is recognizable. Since $L(\mathcal{A}) = \varphi^{-1}(T(\mathcal{A}))$, the characteristic property of recognizable real trace language recalled in Section 2.2 asserts that the language $T(\mathcal{A})$ is recognizable too. Hence we can claim:

**Corollary 3.5.** Let $\mathcal{A}$ be a BACA, then the word language $L(\mathcal{A})$ and the trace language $T(\mathcal{A})$ are both recognizable.

Note that, from Proposition 3.4 and classical results on recognizable word languages, we get that there exist recognizable trace languages which are not accepted by **deterministic** BACA .

## 4. Rational constructions on BACA.

We have seen in the previous section that BACA accept recognizable real trace languages. The aim of this section is, conversely, to construct a BACA for each recognizable real trace language. In order to prove this result, we use the rational expression of these languages recalled in Section 2.2. Hence, we need to study closure properties of the family of BACA under rational operators.
Since we will need this property for the closure under union, we recall that each ACA can be transformed in an equivalent complete ACA [Die90]. An ACA $\mathcal{A} = \langle (Q_a)_{a\in X}, (\delta_a)_{a\in X}\rangle$ is said to be complete if $\delta_a(q_{D(a)}) \neq \emptyset$ for all $a \in X$ and $q_{D(a)} \in Q_{D(a)}$.
The closure under union and intersection of the BACA family will be treated using a cartesian product of automata as for word automata.

**Proposition 4.1.** Let $\mathcal{A}_1$, $\mathcal{A}_2$ be BACA. There exists a BACA $\mathcal{A}$ such that $T(\mathcal{A}) = T(\mathcal{A}_1) \cup T(\mathcal{A}_2)$ (resp. $T(\mathcal{A}) = T(\mathcal{A}_1) \cap T(\mathcal{A}_2)$). Moreover, if $\mathcal{A}_1$, $\mathcal{A}_2$ are deterministic, so is $\mathcal{A}$.

**Proof.** For i = 1,2, consider the BACA $\mathcal{A}_i = \langle (Q_{i,a})_{a\in X}, (\delta_{i,a})_{a\in X}, S_i, F_i, \mathcal{R}_i\rangle$ and let $Q_i = \prod_{a\in X} Q_{i,a}$. Thanks to the canonical bijection between $\prod_{a\in X} (Q_{1,a}\times Q_{2,a})$ and $Q_1\times Q_2$, we identify these two sets throughout this construction.
Define the cartesian product $\mathcal{A} = \mathcal{A}_1\times\mathcal{A}_2 = \langle (Q_a)_{a\in X}, (\delta_a)_{a\in X}, S\rangle$ by:

- $Q_a = Q_{1,a}\times Q_{2,a}$ ,
- $\delta_a((q_1,q_2)_{D(a)}) = \delta_{1,a}((q_1)_{D(a)})\times\delta_{2,a}((q_2)_{D(a)})$
- $S = S_1\times S_2$ .

For i=1,2, we denote $\Pi_i$ the projections defined by $\Pi_i(q_1,q_2) = q_i$ for all $(q_1,q_2) \in Q_1\times Q_2$.
Then, the BACA $\mathcal{A}$ with final states $F_1\times F_2$ and acceptance table $\mathcal{R} = \{R \in Cell(Q_X) / \forall\, i \in \{1,2\}, \Pi_i(R) \supseteq R_i \text{ for some } R_i \in \mathcal{R}_i\}$ accepts the language $T(\mathcal{A}_1) \cap T(\mathcal{A}_2)$.
For the union, as we recalled above, we may assume that $\mathcal{A}_1$ and $\mathcal{A}_2$ are complete. Hence, in a similar way, the BACA $\mathcal{A}$ with final states $F = Q_1\times F_2 \cup F_1\times Q_2$ and acceptance table $\mathcal{R} = \{R \in Cell(Q_X) / \exists\, i \in \{1,2\}, \Pi_i(R) \supseteq R_i \text{ for some } R_i \in \mathcal{R}_i\}$ accepts the language $T(\mathcal{A}_1) \cup T(\mathcal{A}_2)$. ◊

Now, we turn to the concatenation. To this purpose, we first need an auxiliary lemma which allows to clearly distinguish between finite accepted paths and infinite ones.

**Lemma 4.2.** Let $\mathcal{A}$ be a BACA. There exists a BACA $\mathcal{A}'$ such that $T(\mathcal{A}') = T(\mathcal{A})$ and

- for each final state $f=(f_a)_{a\in X}$ of $\mathcal{A}'$, $f_a$ is a dead state for all $a \in X$.
- for each element R of the acceptance table of $\mathcal{A}'$, $R_a$ contains at least two states for some $a \in X$.

**Proof.** Let $\mathcal{A} = \langle (Q_a)_{a\in X}, (\delta_a)_{a\in X}, S, F, \mathcal{R}\rangle$ be a BACA and let $Q = \prod_{a\in X} Q_a$. Intuitively, in order to achieve the first condition we use new final states which are copies of the original ones and from which no transition are possible. For the second condition, we unfold once each elementary loop $\left(p_a \in \delta_a(p_{D(a)})\right)$ which ensure that there is no stationary state in a path and allows to restrict ourselves to tables with at least two states. More formally, the construction follows. Again, we identify the two sets $\prod_{a\in X} (Q_a\times\{-1,0,1\})$ and $Q\times\{-1,0,1\}^X$.

Define the BACA $\mathcal{A}' = \langle (Q'_a)_{a\in X}, (\delta'_a)_{a\in X}, S', F', \mathcal{R}'\rangle$ by:

- $Q'_a = Q_a\times\{-1,0,1\}$ ,
- $(q_a,k_a) \in \delta'_a((p,j)_{D(a)})$ if $j_a \neq 0$ and $q_a \in \delta_a(p_{D(a)})$ and ($k_a = 0$ or $k_a = -j_a$)
- $S' = S\times\{0,1\}^X$ ,
- $F' = F\times\{0\}^X$ and
- $\mathcal{R}' = \{R' \in \mathrm{Cell}(Q'_X) \ / \ \Pi_1(R') \supseteq R$ for some $R \in \mathcal{R}$, $\Pi_2(R') \subseteq \{-1,1\}^X$ and $|R'| \geq 2\}$

where the projections $\Pi_i$ are defined by:

- $\Pi_1(q_a,j_a) = q_a$ and $\Pi_2(q_a,j_a) = j_a$ for all $(q_a,j_a) \in Q'_a$ ,
- $\Pi_1(q,j) = q$ and $\Pi_2(q,j) = j$ for all $(q,j) \in Q\times\{-1,0,1\}^X$ and
- $\Pi_1(P') = q_0,a_1,q_1,a_2,q_2,\ldots$ for all path $P' = (q_0,j_0),a_1,(q_1,j_1),a_2,(q_2,j_2),\ldots$ of $\mathcal{A}'$.

Since the BACA $\mathcal{A}'$ satisfies obviously the required condition, it remains to prove that $T(\mathcal{A}') = T(\mathcal{A})$. Let P' be a path of $\mathcal{A}'$ and let $P = \Pi_1(P')$ then $\ell(P') = \ell(P)$, $\Pi_1(s(P')) = s(P)$ and $\Pi_1(f(P')) = f(P)$. Hence, if P' is an accepted path of $\mathcal{A}'$ then P is an accepted path of $\mathcal{A}$, which prove that $T(\mathcal{A}') \subseteq T(\mathcal{A})$.

Conversely, let $P = q_0,a_1,q_1,\ldots,a_n,q_n$ be a finite accepted path of $\mathcal{A}$. Define inductively $j_0,j_1,\ldots,j_n$ by $\forall\ a \in X$,

$$j_{0,a} = \begin{cases} 0 & \text{if } |a_1\ldots a_n|_a = 0 \\ 1 & \text{otherwise} \end{cases}$$

and $\forall\ 1 \leq k \leq n$,

$$j_{k,a} = \begin{cases} j_{k-1,a} & \text{if } a \neq a_k \\ 0 & \text{if } a = a_k \text{ and } |a_{k+1}\ldots a_n|_a = 0 \\ -j_{k-1,a} & \text{otherwise} \end{cases}$$

We can easily verify that $P' = (q_0,j_0),a_1,(q_1,j_1),\ldots,a_n,(q_n,j_n)$ is an accepted path of $\mathcal{A}'$.

In a similar way, let $P = q_0,a_1,q_1,a_2,q_2,\ldots$ be an infinite accepted path of $\mathcal{A}$. Define inductively $j_0,j_1,j_2,\ldots$ by $j_0 = (1,\ldots,1)$ and $\forall\ 1 \leq k \leq n$, $\forall\ a \in X$,

$$j_{k,a} = \begin{cases} j_{k-1,a} & \text{if } a \neq a_k \\ -j_{k-1,a} & \text{otherwise} \end{cases}$$

We can easily verify that $P' = (q_0,j_0),a_1,(q_1,j_1),a_2,(q_2,j_2),\ldots$ is a path of $\mathcal{A}'$ such that $\Pi_1(P') = P$, whence $s(P') \in S'$. Let us see that $R' = f(P') \in \mathcal{R}'$. By construction, we have $\Pi_2(R') \subseteq \{-1,1\}^X$. Now, $\Pi_1(f(P')) = f(P) \supseteq R$ for some $R \in \mathcal{R}$ since P is accepted. Finally, $\forall\ a \in X$ such that $|\ell(P')|_a = \omega$, we have $|R'_a| \geq 2$ since there is no elementary loop in $\mathcal{A}'$. ◊

Note that in the lemma above, the construction's part which concerns acceptance tables preserves determinism whereas the part for dead final states does not. This is not a problem since we know that the concatenation of two deterministic recognizable word languages may be a non deterministic one. For instance, $(a+b)^*a^\omega$ is non deterministic whereas $(a+b)^*$ and $a^\omega$ are deterministic (see [PP91] for more details about deterministic word languages).

**Proposition 4.3.** Let $\mathcal{A}_1$, $\mathcal{A}_2$ be BACA such that $\mathrm{alphinf}(r_1)\times\mathrm{alph}(r_2) \subseteq I$ for all $(r_1,r_2) \in T(\mathcal{A}_1)\times T(\mathcal{A}_2)$ (in particular, this condition is satisfied if $T(\mathcal{A}_1) \subseteq \mathbb{M}$ or if $T(\mathcal{A}_1)$ and $T(\mathcal{A}_2)$ are fully independent). Then the concatenation $T(\mathcal{A}_1)\cdot T(\mathcal{A}_2)$ is accepted by some BACA.

**Proof.** First, we define the I-shuffle of two word languages $L_1, L_2 \subseteq X^\infty$ by:

$$L_1 \sqcup\!\sqcup_I L_2 = \{u_1v_1u_2v_2\ldots \in X^\infty \,/\, u_1u_2\ldots \in L_1,\ v_1v_2\ldots \in L_1 \text{ with } u_i, v_i \in X^* \text{ for all } i \text{ and } (u_i,v_j) \in I \text{ for all } i > j\}.$$

This operation is the word counterpart of the partial concatenation of real trace languages used in this proposition (cf. [DGP91]). More precisely, let $T_1, T_2 \subseteq \mathbb{R}$ and let $T = \{r\cdot s \in \mathbb{R} \,/\, r \in T_1,\ s \in T_2 \text{ and } \mathrm{alphinf}(r)\times\mathrm{alph}(s) \subseteq I\}$. Then, $\varphi^{-1}(T) = \varphi^{-1}(T_1) \sqcup\!\sqcup_I \varphi^{-1}(T_2)$. Hence, we have to construct a BACA $\mathcal{A}$ such that $L(\mathcal{A}) = L(\mathcal{A}_1) \sqcup\!\sqcup_I L(\mathcal{A}_2)$. This construction is indeed inspired from the I-shuffle operation.

Consider the BACA $\mathcal{A}_i = \langle (Q_{i,a})_{a\in X}, (\delta_{i,a})_{a\in X}, S_i, F_i, \mathcal{R}_i\rangle$ for i=1,2. By Lemma 4.2, we may assume that each final state in $F_i$ is a dead state and that for each element R of the acceptance tables $\mathcal{R}_i$, $R_a$ contains at least two states for some $a \in X$. For all $a \in X$, let $Q_{3,a} = \{1,2\}$. For i=1,2,3, let $Q_i = \prod_{a\in X} Q_{i,a}$. Thanks to the canonical bijection between $\prod_{a\in X} (Q_{1,a}\times Q_{2,a}\times Q_{3,a})$ and $Q_1\times Q_2\times Q_3$, we identify these two sets throughout the proof.

Define the I-shuffle $\mathcal{A} = \mathcal{A}_1 \sqcup\!\sqcup_I \mathcal{A}_2 = \langle (Q_a)_{a\in X}, (\delta_a)_{a\in X}, S, F, \mathcal{R}\rangle$ by:

- $Q_a = Q_{1,a}\times Q_{2,a}\times Q_{3,a}$ ,
- $\delta_a((q_1,q_2,q_3)_{D(a)}) =$
$$\begin{cases} \{q_{1,a}\}\times\delta_{2,a}((q_2)_{D(a)})\times\{2\} & \text{if } q_{3,b} = 2 \text{ for some } b \in D(a) \\ \{q_{1,a}\}\times\delta_{2,a}((q_2)_{D(a)})\times\{2\} \cup \delta_{1,a}((q_1)_{D(a)})\times\{q_{2,a}\}\times\{1\} & \text{otherwise} \end{cases}$$
- $S = S_1\times S_2\times\{\mathbf{1}\}$ where $\mathbf{1}$ stands for the tuple $(1,\ldots,1)$ ,
- $F = F_1\times F_2\times Q_3$ and
- $\mathcal{R} = \{R \in \mathrm{Cell}(Q_X) \,/\, \forall\, i \in \{1,2\},\ \Pi_i(R) \cap F_i \neq \emptyset \text{ or } \Pi_i(R) \supseteq R_i \text{ for some } R_i \in \mathcal{R}_i\}$

where, for i = 1,2,3, the projections $\Pi_i$ are defined by:

- $\Pi_i(q_{1,a},q_{2,a},q_{3,a}) = q_{i,a}$ for all $(q_{1,a},q_{2,a},q_{3,a}) \in Q_a$ and
- $\Pi_i(q_1,q_2,q_3) = q_i$ for all $(q_1,q_2,q_3) \in Q_1\times Q_2\times Q_3$.

For i=1,2, we extend these projections to transitions and paths of $\mathcal{A}$ by:

- let $t = (p_1,p_2,p_3),a,(q_1,q_2,q_3)$ be a transition of $\mathcal{A}$. Then $\Pi_i(t) = \begin{cases} \emptyset & \text{if } q_{3,a} \neq i \\ p_i,a,q_i & \text{otherwise} \end{cases}$
- let $P = t_1t_2\ldots$ be a path of $\mathcal{A}$ and let $J = \{j \geq 1 \,/\, \Pi_i(t_j) \neq \emptyset\}$. Then
$$\Pi_i(P) = \begin{cases} \Pi_i(s(P)) & \text{if } J = \emptyset \\ \Pi_i(t_{n_1})\Pi_i(t_{n_2})\ldots & \text{otherwise where } n_1<n_2<\ldots \text{ is the enumeration of } J \end{cases}$$

Using these definitions, one can prove

**Claim 1:** let P be a path of $\mathcal{A}$. Then $P_1 = \Pi_1(P)$ and $P_2 = \Pi_2(P)$ are paths of $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively and $\ell(P) \in \ell(P_1) \sqcup\!\sqcup_I \ell(P_2)$. Conversely, let $P_1$ and $P_2$ be paths of $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively. For all $w \in \ell(P_1) \sqcup\!\sqcup_I \ell(P_2)$, there exists a unique path P of $\mathcal{A}$ such that $\ell(P) = w$, $P_i = \Pi_i(P)$ for i = 1,2 and $\Pi_3(s(P)) = \mathbf{1}$.

Moreover, it is easy to verify that

**Claim 2:** let P be a path of $\mathcal{A}$. Then for i = 1,2,
- $\Pi_i(s(P)) = s(P_i)$ ,
- $\Pi_i(f(P)) = f(P_i)$ ,

Using these two claims we prove that the BACA $\mathcal{A}$ accepts the language $L(\mathcal{A}_1) \sqcup\!\sqcup_I L(\mathcal{A}_2)$.
Let $w \in L(\mathcal{A}_1) \sqcup\!\sqcup_I L(\mathcal{A}_2)$. For i=1,2, there exists $u_i \in L(\mathcal{A}_i)$ such that $w \in u_1 \sqcup\!\sqcup_I u_2$. Let $P_i$ be a path of $\mathcal{A}_i$ which accepts $u_i$. By Claim 1, there exists a unique path P of $\mathcal{A}$ such that $\ell(P) = w$, $P_i = \Pi_i(P)$ for i = 1,2 and $\Pi_3(s(P)) = \mathbf{1}$. Using Claim 2 we deduce that P is accepted. Hence $w \in L(\mathcal{A})$.
Conversely, let $w \in L(\mathcal{A})$. Consider a path P of $\mathcal{A}$ accepting w. For i=1,2, let $P_i = \Pi_i(P)$. By Claim 1, $P_1$ and $P_2$ are paths of $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively and $w \in \ell(P_1) \sqcup\!\sqcup_D \ell(P_2)$. If P is finite then $P_1$ and $P_2$ are accepted paths (Claim 2). Now assume that P is infinite and let $R \in \mathcal{R}$ be such that $R \subseteq f(P)$.
- Assume that $\Pi_i(R) = \{q_i\}$ is a singleton. Since each element of the acceptance table $R_i \in \mathcal{R}_i$ contains at least two states, we obtain $q_i \in F_i$. Hence $q_i$ is a dead state and $P_i$ must be finite. Using Claim 2, we deduce that $P_i$ is accepted.
- Otherwise, $\Pi_i(R) \supseteq R_i$ for some $R_i \in \mathcal{R}_i$. Since $|\Pi_i(R)| \geq |R_i| \geq 2$, $P_i$ must be infinite. Hence, it is accepted as well since $f(P_i) = \Pi_i(f(P)) \supseteq \Pi_i(R) \supseteq R_i$.

Therefore, for i=1,2, $\ell(P_i) \in L(\mathcal{A}_i)$. Hence $w \in L(\mathcal{A}_1) \sqcup\!\sqcup_I L(\mathcal{A}_2)$. ◊

Note that in order to obtain the main result of this section, we only need the special case of the proposition where the first language of the concatenation is a set of finite traces. For this restricted case, we can also use a construction which is conceptually a bit simpler. Roughly speaking, it is the disjoint union of the automata in which we mix up the final states of the first automata with the initial states of the second one. In fact, these constructions are very close which explains why we present the more general one.

It remains to construct BACA which accepts the infinite iteration of a language accepted by some ACA. Since the infinite iteration of a recognizable language of finite traces is not necessarily recognizable, we need some restriction on the ACA. The following proposition gives the major construction which applies when the ACA has a unique initial state, a unique final state and accepts traces with alphabet X only .

**Proposition 4.4.** Let $\mathcal{A}$ be an ACA with a unique initial state and a unique final state and which verifies alph(t) = X for all $t \in T(\mathcal{A})$. Then $T(\mathcal{A})^\omega$ is accepted by some BACA.

**Proof.** Let $\mathcal{A} = \langle (Q_a)_{a\in X}, (\delta_a)_{a\in X}, \{i\}, \{f\}, \varnothing \rangle$ be an ACA and let $Q = \prod_{a\in X} Q_a$. For $N \geq 2$, let $G_N$ be the cyclic group $\mathbb{Z}/_{N\mathbb{Z}}$ and let $\mathbf{G_N} = (G_N)^X = \prod_{a\in X} G_N$. Once again, thanks to the canonical bijection between $\prod_{a\in X} (Q_a \times G_N)$ and $Q \times \mathbf{G_N}$, we identify these two sets throughout the proof.

For $N \geq 2$, define the BACA $\mathcal{A}_N = \langle (Q'_a)_{a\in X}, (\delta'_a)_{a\in X}, \{i'\}, \varnothing, \mathcal{R} \rangle$ by:

- $Q'_a = Q_a \times G_N$ ,
- $(q'_a, j'_a) \in \delta'_a((q,j)_{D(a)})$ if $\forall\, b \in D(a)$, $(j_b = j'_a)$ or $(j_b = j'_a - 1$ and $q_b = f_b)$ and $q'_a \in \delta_a(p_{D(a)})$ where $p_b = \begin{cases} q_b & \text{if } j_b = j'_a \\ i_b & \text{otherwise} \end{cases}$
- $i' = i \times \{\mathbf{0}\}$ where $\mathbf{0}$ stands for the tuple $(0,\ldots,0)$ of $\mathbf{G_N}$ and
- $\mathcal{R} = \{R \in \mathrm{Cell}(Q'_X) \,/\, \forall\, a \in X,\ \Pi_2(R_a) = G_N\}$ where $\Pi_2(q_a, j_a) = j_a$ for all $(q_a, j_a)$.

First, we prove that $T^\omega \subseteq T(\mathcal{A}_N)$. Let $t \in T^\omega$, we have $t = t_0 t_1 t_2 \ldots$ with $t_m \in T$ for all $m \in \mathbb{N}$. For all $m \in \mathbb{N}$, let $u_m \in \varphi^{-1}(t_m)$ and let $P_m$ be the path of $\mathcal{A}$ which accepts $u_m$. With each $P_m$, we associate a path $P'_m$ of $\mathcal{A}_N$ as follows.

Case m=0: Let $P_0 = q_0, a_1, q_1, \ldots, a_n, q_n$. We set $P'_0 = (q_0,\mathbf{0}), a_1, (q_1,\mathbf{0}), \ldots, a_n, (q_n,\mathbf{0})$. Note that, $P'_0$ is clearly a path of $\mathcal{A}_N$ with $s(P'_0) = (i,\mathbf{0}) = i'$ and $f(P'_0) = (f,\mathbf{0})$.

Case m ≥ 1: Let $P_m = q_0, a_1, q_1, \ldots, a_n, q_n$. We set $P'_m = (q'_0, j_0), a_1, (q'_1, j_1), \ldots, a_n, (q'_n, j_n)$ with $(q'_0, j_0) = (f, \mathbf{m\text{-}1})$ where $\mathbf{m\text{-}1}$ stands for the tuple $(m-1,\ldots,m-1)$ of $\mathbf{G_N}$ and for $1 \leq k \leq n$ and $b \in X$, $(q'_{k,b}, j_{k,b}) = \begin{cases} (q'_{k-1,b}, j_{k-1,b}) & \text{if } b \neq a_k \\ (q_{k,b}, m) & \text{otherwise} \end{cases}$

It is easy to check that for $0 \leq k \leq n$ and $b \in X$, it holds

$$(j_{k,b} = m \text{ and } q'_{k,b} = q_{k,b}) \text{ or } (j_{k,b} = m - 1 \text{ and } q'_{k,b} = f_b \text{ and } q_{k,b} = i_b) \qquad (1)$$

Let $k \in \{1,\ldots,n\}$ and let $a = a_k$, using (1) it can be easily verified that

$$(q'_{k,a}, j_{k,a}) \in \delta'_a((q'_{k-1,b}, j_{k-1,b})_{b\in D(a)})$$

Hence, $P'_m$ is a path of $\mathcal{A}_N$. Moreover, since $\mathrm{alph}(u_m) = \mathrm{alph}(t_m) = X$, we deduce using (1) that $f(P'_m) = (q_n, \mathbf{m}) = (f, \mathbf{m})$.

We have constructed paths $P'_m$ of $\mathcal{A}_N$ such that $f(P'_m) = s(P'_{m+1})$ for all $m \in \mathbb{N}$. Therefore, $P' = P'_0 P'_1 P'_2 \ldots$ is a path of $\mathcal{A}_N$. Let us prove now that P' is accepted. First, P' starts with the initial state since $s(P') = s(P'_0) = i'$. Now, for each $m \geq 1$ and $a \in X$, the counter of the letter a is incremented by 1 in $P'_m$. Since $G_N$ is a finite cyclic group and $Q_a$ is finite as well, it follows that $\Pi_2(f_a(P')) = G_N$. Therefore, $f(P') \in \mathcal{R}$ and P' is accepted. Hence, $t = \varphi(u_0 u_1 u_2 \ldots) = \varphi(\ell(P')) \in T(\mathcal{A}_N)$.

Conversely, we prove that for $N \geq 3$, $T(\mathcal{A}_N) \subseteq T^\omega$. To this purpose, the (infinite) asynchronous cellular automata $\mathcal{A}_\infty = \langle (Q'_a)_{a\in X}, (\delta'_a)_{a\in X}, \{i'\} \rangle$ is defined as $\mathcal{A}_N$ by simply replacing the cyclic group $G_N$ by $\mathbb{N}$.

Let $t \in T(\mathcal{A}_N)$, let $u \in \varphi^{-1}(t)$ and let $P = (q_0, j_0), a_1, (q_1, j_1), a_2, (q_2, j_2) \ldots$ be a path of $\mathcal{A}_N$ which accepts u. We define an associated infinite sequence $P_\infty = (q_0, k_0), a_1, (q_1, k_1), a_2, (q_2, k_2) \ldots$ in $\mathcal{A}_\infty$. We will show in the sequel that $P_\infty$ is in fact a path in $\mathcal{A}_\infty$. The infinite sequence $k_0, k_1, \ldots$ is defined inductively by:

- $k_0 = \mathbf{0} = (0,\ldots,0)$ ,
- for $n \geq 1$ and $a \in X$, $k_{n,a} = \begin{cases} k_{n-1,a} & \text{if } j_{n,a} = j_{n-1,a} \\ 1 + k_{n-1,a} & \text{otherwise } (j_{n,a} = 1 + j_{n-1,a} \text{ in } G_N) \end{cases}$

Fact 1: It is immediate that for all $n \in \mathbb{N}$ and $a \in X$, the image of $k_{n,a}$ in $G_N$ is $j_{n,a}$.

Fact 2: For all $n \in \mathbb{N}$ and $(a,b) \in D$ it holds $|k_{n,b} - k_{n,a}| \leq 1$. We prove this fact by induction. The case $n = 0$ is clear. Let $n \geq 1$ and assume that for all $(a,b) \in D$, $|k_{n-1,b} - k_{n-1,a}| \leq 1$.
Let $(a,b) \in D$, $a \neq b$. Either $k_{n,a} = k_{n-1,a}$ and $k_{n,b} = k_{n-1,b}$ and the result follows from the hypothesis. Otherwise, for instance $a = a_n$ and we have $k_{n,a} = 1 + k_{n-1,a}$ and $k_{n,b} = k_{n-1,b}$. Since $(q_{n-1},j_{n-1}),a,(q_n,j_n)$ is a transition of $\mathcal{A}_N$ we obtain $j_{n,b} = j_{n-1,b} \in \{j_{n-1,a}, 1 + j_{n-1,a}\}$. Since $N \geq 3$, using Fact 1 and the induction hypothesis, we infer $k_{n-1,b} \in \{k_{n-1,a}, 1 + k_{n-1,a}\}$. Therefore, $0 \leq k_{n,a} - k_{n,b} \leq 1$ which conclude the proof of Fact 2.

Fact 3: $P_\infty = (q_0,k_0),a_1,(q_1,k_1),a_2,(q_2,k_2)\ldots$ is a path of $\mathcal{A}_\infty$ with $(q_0,k_0) = (i,\mathbf{0}) = i'$ and no repeated state i.e. $f_a(P_\infty) = \emptyset$ for all $a \in X$.
Let $n \geq 1$ and $a = a_n$. Using Facts 1 and 2 we obtain for all $b \in D(a)$,

$$(k_{n-1,b} = k_{n,a} \Leftrightarrow j_{n-1,b} = j_{n,a}) \text{ and } (k_{n-1,b} = k_{n,a} - 1 \Leftrightarrow j_{n-1,b} = j_{n,a} - 1) \qquad (2)$$

Since $(q_{n-1},j_{n-1}),a,(q_n,j_n)$ is a transition of $\mathcal{A}_N$ we obtain using (2) that $(q_{n-1},k_{n-1}),a,(q_n,k_n)$ is a transition of $\mathcal{A}_\infty$. Therefore, $P_\infty$ is a path of $\mathcal{A}_\infty$. Since P is accepted, $q_0 = i$ and $s(P_\infty) = (i,\mathbf{0})$. Moreover, for all $a \in X$, $f_a(P) = G_N$. Hence, the counter of a is increased infinitely often in P and in $P_\infty$ which prove that there is no repeated state in $P_\infty$.

Let $m \in \mathbb{N}$ and $t = (q,k),a,(q',k')$ be a transition of $P_\infty$. We define

$$\Pi_m(t) = \begin{cases} \emptyset & \text{if } k'_a \neq m \\ \alpha,a,\alpha' & \text{otherwise } (k'_a = m) \text{ with} \end{cases}$$

$$\alpha'_a = q'_a,\ \alpha_a = \begin{cases} i_a & \text{if } k_a = m-1 \\ q_a & \text{if } k_a = m \end{cases} \quad \text{and} \quad \forall\, b \neq a,\ \alpha'_b = \alpha_b = \begin{cases} i_b & \text{if } k_b < m \\ q_b & \text{if } k_b = m \\ f_b & \text{if } k_b > m \end{cases}$$

Fact 4: If $\Pi_m(t) \neq \emptyset$ then it is a transition of $\mathcal{A}$.
Since t is a transition of $\mathcal{A}_\infty$ we have $(q'_a,k'_a) \in \delta'_a((q_b,k_b)_{b\in D(a)})$. Hence, $\forall\, b \in D(a)$, $(k_b = k'_a)$ or $(k_b = k'_a - 1$ and $q_b = f_b)$ and $q'_a \in \delta_a((p_b)_{b\in D(a)})$ where $p_b = \begin{cases} q_b & \text{if } k_b = k'_a \\ i_b & \text{otherwise} \end{cases}$.
Therefore, $\alpha_b = p_b$ for all $b \in D(a)$ and then $\alpha'_a = q'_a \in \delta_a((\alpha_b)_{b\in D(a)})$.

Fact 5: Assume that $P_\infty = P'sP''tP'''$ with $\Pi_m(s) = \alpha,a,\alpha'$, $\Pi_m(t) = \beta,b,\beta'$ and $\Pi_m(P'') = \emptyset$. Then $\alpha' = \beta$.
Let $s = (p,h),a,(p',h')$, let $t = (q,k),b,(q',k')$ and let $c \in X$.

- Case $h'_c > m$: We have $k_c \geq h'_c > m$ and then $\alpha'_c = f_c = \beta_c$.
- Case $h'_c < m$: Since $\Pi_m(P'') = \emptyset$, we must have $k_c < m$. Hence, $\alpha'_c = i_c = \beta_c$.
- Case $h'_c = m$: Assume that there is no transition with label c in P''. Then, we have $(p'_c,h'_c) = (q_c,k_c)$. Hence, $\alpha'_c = p'_c = q_c = \beta_c$. Otherwise, let $(r,n),c,(r',n')$ be the first transition with label c in P''. We have $(p'_c,h'_c) = (r_c,n_c)$. Since $\Pi_m(P'') = \emptyset$, we must have $n'_c \neq m$. Hence $n'_c = n_c + 1$ and $r_c = f_c$. Therefore, $\alpha'_c = p'_c = f_c = \beta_c$.

Fact 6: Assume that $P_\infty = P'tP''$ with $\Pi_m(t) = \alpha,a,\alpha'$. Then

$$\Pi_m(P') = \emptyset \Rightarrow \alpha = i \qquad \text{and} \qquad \Pi_m(P'') = \emptyset \Rightarrow \alpha' = f.$$

Let $t = (q,k),a,(q',k')$ and let $b \in X$. If $\Pi_m(P') = \emptyset$ then $k_b < m$ and $\alpha_b = i_b$. If $\Pi_m(P'') = \emptyset$ then $k'_b \geq m$. Either $k'_b > m$ and we have $\alpha'_b = f_b$ or $k'_b = m$. In the last case, since (Fact 3)

$r_b(P_\infty) = \emptyset$, there exist transitions with label b in P''. Let (p,h),b,(p',h') be the first transition with label b in P''. We have $(q'_b,k'_b) = (p_b,h_b)$. Since $\Pi_m(P'') = \emptyset$, we must have $h'_b \neq m$. Hence $h'_b = h_b + 1$ and $p_b = f_b$. Therefore, $\alpha'_b = q'_b = f_b$.

From Facts 4 and 5, it follows that $P_m = \Pi_m(P_\infty)$ is a path of $\mathcal{A}$. Since there is no repeated state in $P_\infty$ (Fact 3), the paths $P_m$ are finite for all m. Moreover, by Fact 6, for each $m \in \mathbb{N}$, the path $P_m$ is accepted. Hence, $u_m = \ell(P_m) \in L(\mathcal{A})$.
In order to conclude the proof, we need the
Fact 7: Assume that $P_\infty = P'(p,h),a,(p',h')P''(q,k),b,(q',k')P'''$. Then $k'_b < h'_a \Rightarrow (a,b) \in I$.
Indeed, if $(a,b) \in D$ then by definition of $\mathcal{A}_\infty$ we have $k_a \leq k'_b$. Hence, $h'_a \leq k_a \leq k'_b$.

Recall that $u = \ell(P) = \ell(P_\infty)$. We claim that $u \sim u_0u_1u_2\ldots$ which conclude the proof. In order to prove the claim, we use the characterization of the equivalence relation $\sim$ by means of prefixes (see Section 2.1).
Let v be a finite prefix of $u_0u_1u_2\ldots$ and let $m \in \mathbb{N}$ be such that $v < u_0\ldots u_m$. Since $P_0,\ldots,P_m$ are finite, we can write $P_\infty = P'P''$ with P' finite and $\Pi_k(P'') = \emptyset$ for all $k \leq m$. Let x (y resp.) be the subword of $\ell(P')$ which consists of the labels of the transitions t of P' such that $\Pi_k(t) \neq \emptyset$ for some $k \leq m$ ($\Pi_k(t) = \emptyset$ for all $k \leq m$ resp.). Using Fact 7, we can easily prove that $\ell(P') \sim xy$. Since $\Pi_k(P'') = \emptyset$ for all $k \leq m$, we also deduce from Fact 7 that $x \sim u_0\ldots u_m$. Therefore, $[v] \leq [u_0\ldots u_m] \leq [\ell(P')]$.

Conversely, let v be a finite prefix of u and let P' be the prefix of $P_\infty$ such that $v = \ell(P')$. Since P' is finite, we can consider $m = \max\{n \in \mathbb{N} / \Pi_n(P') \neq \emptyset\}$. Using the previous case, there exists $y \in X^*$ such that $u_0\ldots u_my$ is equivalent to some finite prefix w of u. From the definition of m, we deduce that $|v|_a \leq |u_0\ldots u_m|_a$ for all $a \in X$. Hence $w = vz$ for some $z \in X^*$. Therefore, $[u_0\ldots u_my] = [vz]$ and using Levi's lemma we infer that $[v] \leq [u_0\ldots u_m]$. ◊

Using all the results above, we obtain the following theorem which admits as a trivial corollary one major result of [GPZ91]: if $T^+$ is a recognizable language of finite traces then $T^\omega$ is recognizable as well.

**Theorem 4.5.** Let $T \in \mathrm{Rec}\ \mathbb{M}$ such that $T = T^+$. Then, there exists a BACA which accepts $T^\omega$.

**Proof.** In order to apply Proposition 4.4, we start with some algebraic decompositions of T. First, we use an easy decomposition [GPZ91] : $T^\omega = \bigcup_{Y \subseteq X} T \cdot T_Y^\omega$ where for all $Y \subseteq X$, $T_Y = \{t \in T / \mathrm{alph}(t) = Y\}$. Using Zielonka's theorem, Propositions 4.1 and Proposition 4.3, it remains to prove that for each $Y \subseteq X$, $T_Y^\omega$ is accepted by some BACA.
Let $D' = D \cap (Y \times Y)$ and let $\mathcal{A} = \langle (Q_a)_{a \in Y}, (\delta_a)_{a \in Y}, \{i\}, F, \emptyset \rangle$ be a deterministic ACA over (Y,D') which accepts $T_Y$ (Zielonka's theorem). For $t \in T_Y$ we denote by $f(t)$ the final state of the path which accepts t. For $f \in F$, let $T_f = \{t \in T_Y / f(t) = f\}$. We claim that

$$T_Y^\omega = \bigcup_{f \in F} T_Y \cdot T_f^\omega.$$

One inclusion is clear. Conversely, let $t \in T_Y^\omega$, we have $t = t_1t_2\ldots$ with $t_n \in T_Y$ for all $n \geq 1$. For all pair $\{i,j\} \subseteq \mathbb{N}$ with $i < j$, let $\chi(i,j) = f(t_{i+1}\ldots t_j)$ which is a state of F since $T_Y^+ = T_Y$. By

Ramsey's theorem [Gra81], there exists an infinite subset J of $\mathbb{N}$ and a state f of F such that for all pair $\{i,j\} \subseteq J$ with $i < j$, $\chi(i,j) = f$. Let $i_0 < i_1 < i_2 < \dots$ be the enumeration of J. Then $t_1 \dots t_{i_1} \in T_Y^+ = T_Y$ and for all $k \geq 1$, $t_{1+i_k} \dots t_{i_{k+1}} \in T_f$. Therefore, $t = t_1 t_2 \dots \in T_Y \cdot T_f^\omega$ and the claim is proved.
By Proposition 4.4, $T_f^\omega$ is accepted by some BACA over (Y,D'). Using a similar construction than in the finite case, it is easy to transform this BACA into a BACA over (X,D) which accepts $T_f^\omega$. We conclude the proof using Proposition 4.1 and Proposition 4.3. ◊

As a consequence of the constructions for union, iteration and infinite iteration, we obtain:

**Theorem 4.6.** The family of languages accepted by BACA is exactly Rec $\mathbb{R}$.

**Proof.** We have seen in Section 3 that if $\mathcal{A}$ is a BACA then $T(\mathcal{A})$ is recognizable. Conversely, let $R \in$ Rec $\mathbb{R}$ and let $\eta : \mathbb{M} \to S$ be a morphism which recognizes R. For $s \in S$, let $R_s = \eta^{-1}(s)$. We have recalled in Section 2.2 that

$$R = \bigcup_{(s,e) \in P} R_s R_e^\omega \text{ where } P = \{(s,e) \in S^2 \,/\, se=s,\ e^2=e \text{ and } R_s R_e^\omega \subseteq T\}.$$

For all $s \in S$, $R_s \in$ Rec $\mathbb{M}$ and is accepted by some ACA (Zielonka's theorem). Now, if $e \in S$ is an idempotent ($e^2=e$) then $R_e = R_e^+$. Hence, $R_e^\omega$ is accepted by some BACA (Theorem 4.5). The result follows from Proposition 4.1 and Proposition 4.3. ◊

Note that, in the constructions of Theorems 4.5 and 4.6, we could use for recognizable languages of finite traces non deterministic asynchronous cellular automata which are much more easy to obtain than deterministic ones (see [Pig92]).

## 5. Conclusion

In this paper we have proved that a real trace language is recognizable if and only if it is recognized by a Büchi asynchronous cellular automaton. The theory presented in Section 3 can easily be extended to define the notion of Büchi asynchronous automata. Then a very similar construction than in the finite case shows that a BACA can be transformed in a Büchi asynchronous automaton recognizing the same language. Hence, as BACA, Büchi asynchronous automata accept exactly the recognizable real trace languages.
Our main result is a generalization of the deep Zielonka's theorem to infinite real trace languages. This constitutes the main part of the extension of this theorem to complex traces. This complete extension is not presented here for lack of space. Note that Büchi asynchronous (cellular) automata can recognize real traces only since the label of an infinite path is a real trace. Therefore, to generalize the theory to complex traces it is necessary to transform these automata so that they can take into account the second alphabetic component of a complex trace.
The main drawback of Büchi automata is that non deterministic automata are needed to accept all recognizable languages. On the contrary, in the free monoid $X^\infty$, a language is recognizable if and only if it is accepted by some deterministic Muller automaton. All the results presented in Section 3 can be easily extended to define the proper classes of Muller asynchronous (cellular) automata. Hence, a major open problem is the characterization of recognizable infinite trace languages by such automata.

**Acknowledgement:** *We thank V. Diekert, H.J. Hoogeboom and W. Zielonka for fruitful discussions.*

## 6. References

[AR88] I.J. AALBERSBERG and G. ROZENBERG, "Theory of traces", Theoretical Computer Science 60, p. 1-82, 1988.

[Ber79] J. BERSTEL, "Transductions and context free languages", Teubner Studienbucher Informatik, 1979.

[BMP89] P. BONIZZONI, G. MAURI and G. PIGHIZZINI, "About infinite traces", Proceedings of the ASMICS Workshop on Partially Commutative Monoids, Tech. Rep. TUM-I 9002, Technische Universität München, 1989.

[CMZ89] R. CORI, Y. METIVIER and W. ZIELONKA, "Asynchronous mappings and asynchronous cellular automata", Tech. Rep. 89-97, LaBRI, Université de Bordeaux, France, 1989.

[Die90] V. DIEKERT, "Combinatorics on traces", Lecture Notes in Computer Science 454, 1990.

[Die91] V. DIEKERT, "On the concatenation of infinite traces", STACS'91, Lecture Notes in Computer Science 480, p. 105-117, 1991.

[DGP91] V. DIEKERT, P. GASTIN, A. PETIT, "Recognizable complex trace languages", MFCS'91, Lecture Notes in Computer Science 520, p. 131-140, 1991.

[Eil74] S. EILENBERG, "Automata, Languages and Machines", Academic Press, New York, 1974.

[Gas90] P. GASTIN, "Infinite traces", Proceedings of the Spring School of Theoretical Computer Science on "Semantics of concurrency", Lecture Notes in Computer Science 469, p. 277-308, 1990.

[Gas91] P. GASTIN, "Recognizable and rational languages of finite and infinite traces", STACS'91, Lecture Notes in Computer Science 480, p. 89-104, 1991.

[GP91] P. GASTIN and A. PETIT, "Büchi (asynchronous) automata for infinite traces", Technical report , LRI, Université Paris-Sud, France, 1991.

[GPZ91] P. GASTIN, A. PETIT, W. ZIELONKA, "A Kleene theorem for infinite trace languages", ICALP'91, Lecture Notes in Computer Science 510, p. 254-266,1991.

[GR91] P. GASTIN and B.ROZOY, "The Poset of infinitary traces", to appear in Theoretical Computer Science, Tech. Rep. 91-07, LITP, Université Paris 6, France, 1991.

[Gra81] R. L. GRAHAM, "Rudiments of Ramsey theory ", Regional conference series in mathematics 45, 1981.

[Kwi90] M.Z. KWIATKOWSKA, "A metric for traces", Information and Processing Letters 35, p. 129-135, 1990.

[Maz77] A. MAZURKIEWICZ, "Concurrent program Schemes and their interpretations", Aarhus University, DAIMI Rep. PB 78, 1977.

[Maz86] A. MAZURKIEWICZ, "Trace theory", Advanced Course on Petri Nets, Lecture Notes in Computer Science 255, p. 279-324, 1986.

[Och85] E. OCHMANSKI, "Regular behaviour of concurrent systems", Bulletin of EATCS 27, p. 56-67, October 1985.

[Per89] D. PERRIN, “Partial commutations”, ICALP’89, Lecture Notes in Computer Science 372, p. 637-651, 1989.

[PP91] D.PERRIN and J.E. PIN, “Mots Infinis”, Tech. Rep. 91-06, LITP, Université Paris 6, France, 1991. Book to appear.

[Pig92] G. PIGHIZZINI, “On the iteration of trace languages”, Proceed. of the ASMICS Workshop on Infinite Traces, to appear as Tech. Rep., Universität Stuttgart, 1992.

[Tho89] W. THOMAS, “On logical definability of trace languages”, Proceedings of the ASMICS Workshop on Partially Commutative Monoids, Tech. Rep. TUM-I 9002, Technische Universität München, p. 172-182, 1989.

[Tho90] W. THOMAS, “Automata on infinite objects”, Handbook of Theoretical Computer Science (J.V. Leeuwen, Ed.), Elsevier, Amsterdam, Vol. B, p. 135-191, 1990.

[Tho91] W. THOMAS, “On logics, tilings, and automata”, ICALP’91, Lecture Notes in Computer Science 510, p. 441-454, 1991.

[Zie87] W. ZIELONKA, “Notes on finite asynchronous automata and trace languages”, RAIRO Theoretical Informatics and Applications 21, p. 99-135, 1987.

[Zi 89] W. ZIELONKA, “Safe execution of recognizable trace languages by asynchronous automata”, Lecture Notes in Computer Science 363, p. 278-289, 1989.