# Deterministic asynchronous automata for infinite traces*

EXTENDED ABSTRACT

Volker Diekert Anca Muscholl

Universität Stuttgart
Institut für Informatik
Breitwiesenstr. 20-22
D-7000 Stuttgart 80

**Abstract.** This paper shows the equivalence between the family of recognizable languages over infinite traces and deterministic asynchronous cellular Muller automata. We thus give a proper generalization of McNaughton's Theorem from infinite words to infinite traces. Thereby we solve one of the main open problems in this field. As a special case we obtain that every closed (w.r.t. the independence relation) word language is accepted by some $I$-diamond deterministic Muller automaton. We also determine the complexity of deciding whether a deterministic $I$-diamond Muller automaton accepts a closed language.

## 1 Introduction

The main result of the present paper provides a positive answer to the open question of [GP92] and extends McNaughton's Theorem to recognizable real trace languages. Concretely, we show the equivalence between the family of recognizable real trace languages and the family of real trace languages which can be accepted by deterministic asynchronous cellular Muller automata. In fact, all our results have a natural extension to complex traces. This is not done here for sake of simplicity and will be done elsewhere.

The paper is organized as follows. Section 2 provides some basic notions. We also recall some facts about recognizable infinitary word and trace languages. Section 3 gives first a technical lemma which allows to follow the approach of Perrin/Pin [PP91]to McNaughton's Theorem. We define deterministic real trace languages and show as the main result of this section that the family of recognizable real trace languages is equivalent to the Boolean closure of deterministic languages. The results we obtain are a proper generalization of the analogous results in the word case. Based on the foregoing result, we show in Section 4 that the family of recognizable real trace languages coincides with the family of languages accepted by deterministic asynchronous cellular Muller automata. In fact, this kind of automaton is a special form of so-called $I$-diamond automata, which

* This research has been supported by the ESPRIT Basic Research Action No. 6317 ASMICS 2.

play an important role in the case of finite traces. In Section 5 we consider the complexity of deciding whether a Muller $I$-diamond automaton accepts a closed language or not. The complete characterization and decision procedure are left to a forthcoming paper.

## 2 Preliminaries

### 2.1 Basic Notions

We denote by $(\Sigma, D)$ a finite *dependence alphabet*, with $\Sigma$ being a finite alphabet and $D \subseteq \Sigma \times \Sigma$ a reflexive and symmetric relation called *dependence relation.* The complementary relation $I = (\Sigma \times \Sigma) \setminus D$ is called *independence relation.* Let also $D(a) = \{b \in \Sigma \mid (a,b) \in D\}$.
The monoid of *finite traces*, $\mathbb{M}(\Sigma, D)$, is defined as a quotient monoid with respect to the congruence relation induced by $I$, i.e., $\mathbb{M}(\Sigma, D) = \Sigma^*/\{ab = ba \mid (a,b) \in I\}$. Traces can be identified with their dependence graph, i.e., with (isomorphism classes of) labelled, acyclic, directed graphs $[V, E, \lambda]$, where $V$ is a set of vertices labelled by $\lambda : V \to \Sigma$ and $E$ is a set of edges with the property that $E \cup E^{-1} \cup \mathrm{id}_V = \lambda^{-1}(D)$. This notion can be extended to infinite dependence graphs. We denote by $\mathbb{G}(\Sigma, D)$ the set of infinite dependence graphs with a countable set of vertices $V$ such that $\lambda^{-1}(a)$ is well-ordered for all $a \in \Sigma$. The requirement that any subset of vertices with the same label should be well-ordered, allows to represent the vertices as pairs $(a, i)$, with $a \in \Sigma$ and $i$ a countable ordinal. Note that $\mathbb{G}(\Sigma, D)$ is a monoid with respect to 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 edges $(v_1, v_2) \in V_1 \times V_2$, whenever $(\lambda_1(v_1), \lambda_2(v_2)) \in D$ holds. The identity is the empty graph $[\emptyset, \emptyset, \emptyset]$. The concatenation is immediately extendable to infinite products. Let $(g_n)_{n\geq 0} \subseteq \mathbb{G}(\Sigma, D)$, then $g = g_0 g_1 \ldots \in \mathbb{G}(\Sigma, D)$ is defined as the disjoint union of the $g_n$, together with new edges from $g_n$ to $g_m$ for $n < m$ between vertices with dependent labels. Thus, we can now define for any $L \subseteq \mathbb{G}(\Sigma, D)$ the $\omega$-iteration as $L^\omega = \{ g_0 g_1 \ldots \mid g_n \in L,\ \forall n \geq 0 \}$.
We denote by $\Sigma^\omega$ the set of infinite words over the alphabet $\Sigma$ (i.e., mappings from $\mathbb{N}$ to $\Sigma$), and by $\Sigma^\infty$ the set of all words $\Sigma^* \cup \Sigma^\omega$. The canonical mapping $\varphi : \Sigma^* \to \mathbb{M}(\Sigma, D)$ can be extended to $\Sigma^\infty$, i.e., $\varphi : \Sigma^\infty \to \mathbb{G}(\Sigma, D)$. The image $\varphi(\Sigma^\infty) \subseteq \mathbb{G}(\Sigma, D)$ is called the set of *real traces* and is denoted by $\mathbb{R}(\Sigma, D)$. In other words, real traces can be identified with (in)finite graphs where every vertex has finitely many predecessors. Observe that $\mathbb{R}(\Sigma, D)$ is not a submonoid of $\mathbb{G}(\Sigma, D)$, since, in general, $\varphi$ commutes neither with concatenation nor with $\omega$-iteration (if $L, K \in \Sigma^\infty$, then $\varphi(LK) = \varphi(L)\varphi(K)$ and $\varphi(L^\omega) = (\varphi(L))^\omega$ hold if and only if $L$ does not contain any infinite word). In the following we will denote $\mathbb{R}(\Sigma, D)$ ($\mathbb{M}(\Sigma, D)$ respectively) by $\mathbb{R}$ ($\mathbb{M}$ respectively). A word language $L \subseteq \Sigma^\infty$ is said to be *closed* (with respect to $(\Sigma, D)$) if $L = \varphi^{-1}\varphi(L)$ for the canonical mapping $\varphi : \Sigma^\infty \to \mathbb{R}$.
Let us give now some further basic notations related to finite traces. For $t \in \mathbb{M}$, we denote by $\mathrm{alph}(t)$ the set of letters occurring in $t$, and by $|t|_a$, $a \in \Sigma$, the

number of $a$ occurring in $t$. Moreover, $\mathrm{alph}(L) = \bigcup_{t \in L} \mathrm{alph}(t)$. We shall often use the abbreviation $(t, u) \in I$ for $\mathrm{alph}(t) \times \mathrm{alph}(u) \subseteq I$. The set of maximal elements of a finite trace $t \in \mathbb{M}$, denoted by $\max(t)$, is defined as the set of labels corresponding to the maximal vertices of the dependence graph of $t$, i.e., $\max(t) = \{a \in \Sigma \mid \exists x \in \Sigma^* : \varphi(xa) = t\}$.
For (in)finite traces the prefix order $\leq$ is given by $u \leq t$ if and only if there exists a trace $s$ with $t = us$. We shall often make use of intersections of prefixes of a given (in)finite trace in the following way: if $t \in \mathbb{R}$, $u \leq t$ and $v \leq t$, then $(u \sqcap v) \leq t$, too. Hereby is $u \sqcap v$ the trace obtained by intersecting the dependence graphs of $u$ and $v$ within the dependence graph of $t$, where the intersection is meant with respect to the labelled vertex sets of $u$, $v$. We use in the same sense the greatest lower (least upper) bound $\sqcap$ ($\sqcup$) of a set $X \subseteq \mathbb{R}$, whenever it exists.

### 2.2 Recognizable word languages

The family of *recognizable* infinitary word languages, denoted by $\mathrm{Rec}(\Sigma^\infty)$, can be defined by means of finite state automata with suitable acceptance conditions for infinite words (see [Tho90]).
Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be a nondeterministic finite automaton where $Q$ is the finite set of states, $\delta \subseteq Q \times \Sigma \times Q$ is the transition relation, $q_0$ is the initial state and $F$ is the set of accepting states. If $\pi$ denotes an infinite path in $\mathcal{A}$, then $\inf(\pi)$ is the set of states occurring infinitely often in $\pi$. An infinite word $w$ is accepted by $\mathcal{A}$ as a Büchi automaton if there exists a transition path $\pi$ of $\mathcal{A}$ labelled by $w$ such that for some final state $f \in F$, $f \in \inf(\pi)$. For the Muller acceptance condition, we have additionally a state table $\mathcal{T} \subseteq \mathcal{P}(Q)$. An infinite word $w$ is accepted by $\mathcal{A}$ as a Muller automaton if there is a path $\pi$ labelled by $w$ such that for some $T \in \mathcal{T}$, $\inf(\pi) = T$. Finite words are accepted by $\mathcal{A}$ in the usual way, by reaching a final state of $F$.
A language $L \subseteq \Sigma^\infty$ is called *recognizable* if it is accepted by some nondeterministic Büchi automaton. Furthermore, the well-known theorem of McNaughton [McN66] states that $\mathrm{Rec}(\Sigma^\infty)$ coincides with the family of infinitary word languages accepted by deterministic Muller automata (see e.g. [Tho90]).

### 2.3 Recognizable Real Trace Languages

One possible way to define *recognizable* infinitary trace languages, denoted by $\mathrm{Rec}(\mathbb{R})$, is by recognizing morphisms [Gas91]. Let $\eta : \mathbb{M} \to S$ be a morphism to a finite monoid $S$. A real trace language $L \subseteq \mathbb{R}$ is *recognized* by $\eta$ if for any sequence $(t_n)_{n \geq 0} \subseteq \mathbb{M}$ the following holds:

$$t_0 t_1 t_2 \ldots \in L \quad \Longrightarrow \quad \eta^{-1}\eta(t_0)\, \eta^{-1}\eta(t_1)\, \eta^{-1}\eta(t_2) \ldots \subseteq L$$

Moreover, in this case $L = \bigcup_{(s,e) \in P} \eta^{-1}(s)\, \eta^{-1}(e)^\omega$ with

$$P = \{\, (s, e) \in S^2 \mid se = s, e^2 = e \text{ and } \eta^{-1}(s)\, \eta^{-1}(e)^\omega \cap L \neq \emptyset \,\}.$$

From the point of view of automata, the family of recognizable real trace languages Rec($\mathbb{R}$) consists of those languages, which can be accepted by a nondeterministic Büchi asynchronous cellular automaton [GP92]. Hereto, we give the definition below.

For the family of finitary recognizable trace languages, Zielonka introduced asynchronous and asynchronous cellular automata ([Zie87], [Zie89]) and he showed the deep result of the equivalence between Rec($\mathbb{M}$) and the family of finitary trace languages accepted by some deterministic asynchronous cellular automaton (or deterministic asynchronous automaton).

An *asynchronous cellular automaton* is a tuple $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, F)$ where for each $a \in \Sigma$, $Q_a$ is a finite set of states, $q_0 \in \prod_{a\in\Sigma} Q_a$ is the initial state, $F \subseteq \prod_{a\in\Sigma} Q_a$ is the set of final states and $\delta_a \subseteq (\prod_{b\in D(a)} Q_b) \times Q_a$ is the local transition relation. We will denote in the following $\prod_{b\in A} Q_b$ ($(q_b)_{b\in A}$, resp.) by $Q_A$ ($q_A$, resp.), where $A \subseteq \Sigma$. In particular, for $a \in \Sigma$ we mean by $q_{D(a)}$ the local states tuple $(q_b)_{b\in D(a)}$.

The global transition relation $\Delta \subseteq Q_\Sigma \times \Sigma \times Q_\Sigma$ of $\mathcal{A}$ is defined by:

$$q' \in \Delta(q,a) \quad \Leftrightarrow \quad (\ q'_a \in \delta_a(q_{D(a)}) \quad \text{and} \quad q'_c = q_c, \text{ for all } c \neq a\ )$$

This means for a global state $q$ and $a \in \Sigma$ that a next state $q' \in \Delta(q,a)$ exists if and only if $\delta_a(q_{D(a)})$ is not empty and in this case, the action of $a$ changes merely the $a$-component of $q$ and the new value $q'_a$ depends only on the $b$ components of $q$, for $b \in D(a)$. Thus, asynchronous cellular automata have the ability of parallel execution of independent actions. Note that this is a typical situation where common read is allowed, whereas writing operations are exclusive and every processor has his own writing domain (CROW).

In [GP92] acceptance conditions have been defined for infinite traces. If we consider an infinite path $\pi = (q_0, a_0, q_1, a_1, \ldots)$ in $\mathcal{A}$, with $q_n \in Q_\Sigma$ and $a_n \in \Sigma$ for every $n \geq 0$, then let $\inf_a(\pi) = \{\, q_a \in Q_a \,|\, \exists^\infty n \text{ s.t. } q_{n,a} = q_a \,\}$ denote the set of local $a$-states which occur infinitely often in $\pi$. Then, an asynchronous cellular automaton viewed as a Büchi (Muller, resp.) automaton is given a table $\mathcal{T} \subseteq \prod_{a\in\Sigma} \mathcal{P}(Q_a)$ i.e., $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, F, \mathcal{T})$. An infinite trace $t \in \mathbb{R}$ is accepted by the Büchi (Muller, respectively) automaton if there exists a path $\pi$ labelled by $t$ (i.e., by any $w \in \varphi^{-1}(t)$) and a set $T \in \mathcal{T}$ such that $\inf_a(\pi) \supseteq T_a$ ($\inf_a(\pi) = T_a$, respectively), for every $a \in \Sigma$. Note that the acceptance conditions are inherently local, thus being appropriate for this kind of automata with decentralized control. Finite traces are accepted in the usual way, by reaching a final state. Gastin/Petit showed the equivalence between Rec($\mathbb{R}$) and the family of real trace languages which are accepted by some nondeterministic Büchi asynchronous cellular automaton (an analogous result holds for Büchi asynchronous automata) [GP92]. However the construction of Gastin/Petit was inherently nondeterministic and so far there is no way to modify their approach (even considering different acceptance conditions) in order to obtain a deterministic automaton. The construction below is based on a totally different approach.

## 3 Algebraic Results on Recognizable Real Trace Languages

Let $L \in \mathrm{Rec}(\mathbb{R})$ be recognized by a morphism $\eta : \mathbb{M} \to S$ to a finite monoid $S$. We define:

$$\mathbb{M}_s = \eta^{-1}(s), \qquad \text{for } s \in S$$
$$\mathbb{P}_s = \mathbb{M}_s \setminus \mathbb{M}_s\mathbb{M}_+, \text{ with } \mathbb{M}_+ = \mathbb{M} \setminus \{1\}$$

Thus, $\mathbb{M}_s$ is the set of all finite traces mapped by $\eta$ to $s \in S$ and $\mathbb{P}_s$ is the subset of $\mathbb{M}_s$ having no proper prefix in $\mathbb{M}_s$. Moreover, we may assume that $\mathrm{alph}(t) = \mathrm{alph}(t')$ for all $t, t' \in \mathbb{M}$ with $\eta(t) = \eta(t')$, since we may replace $S$ by $S \times \mathcal{P}(\Sigma)$, with the multiplication defined by $(s, A)(s', A') = (ss', A \cup A')$ and $(1, \emptyset)$ as neutral element. Hence, $\mathrm{alph}(s)$ for $s \in S$ is well-defined. It can also be viewed as an abbreviation for $\mathrm{alph}(\mathbb{M}_s)$.
From the theory of finite monoids we will need only basic tools, such as the quasi-order relation $\leq_{\mathcal{R}}$ defined as $a \leq_{\mathcal{R}} b$ if and only if $aS \subseteq bS$ and the equivalence relation $\mathcal{R}$ defined as $a\mathcal{R}b$ if and only if$aS = bS$. Furthermore, let $E(S)$ denote the set of idempotent elements of $S$, $E(S) = \{ e \in S \mid e = e^2 \}$ and define the following partial order relation on $E(S)$: $f \leq e$ if and only if $ef = f$ (in fact, $f \leq e$ is equivalent to $f \leq_{\mathcal{R}} e$). By $f < e$ we mean $f \leq e$ and $e \not\leq f$.
Back to infinite traces, the following notation will be used frequently:

$$\mathrm{Inf}(A) = \{ t \in \mathbb{R} \mid \mathrm{alphinf}(t) = A \},$$
$$\mathbb{R}_A = \{ t \in \mathbb{R} \mid D(\mathrm{alphinf}(t)) = D(A) \} \text{ for } A \subseteq \Sigma.$$

where for $t \in \mathbb{R}$, $\mathrm{alphinf}(t)$ denotes the set of letters occurring infinitely often in $t$. In particular, we denote by $\mathrm{Inf}(s)$ and $\mathbb{R}_s$ for $s \in S$ the sets $\mathrm{Inf}(A)$ and $\mathbb{R}_A$ with $A = \mathrm{alph}(s)$.

*Remark 1.* Note that in the word case (i.e., $D = \Sigma \times \Sigma$) we have $\mathbb{R}_A = \Sigma^\omega$, for every $\emptyset \neq A \subseteq \Sigma$ and $\mathbb{R}_\emptyset = \Sigma^*$. We prefer to use $\mathbb{R}_s$ instead of $\mathrm{Inf}(s)$, since thereby we obtain the analogous results for infinite words as a special case of our results.

**Definition 2.** Let $L \subseteq \mathbb{M}$.
We define $\overrightarrow{L} := \{ t \in \mathbb{R} \mid t = \sqcup Y \text{ with } Y \text{ directed and } Y \subseteq L \}$.
(A set $Y \subseteq \mathbb{M}$ is called directed if for every $t, t' \in Y$, there exists an upper bound which also belongs to $Y$.)
In analogy to infinitary word languages, a *deterministic* real trace language is a language of type $\overrightarrow{L} \cap \mathbb{R}_A$, where $L \in \mathrm{Rec}(\mathbb{M})$ and $A \subseteq \Sigma$.

*Remark 3.* i) Note that every deterministic language is recognizable. Clearly, $\mathbb{R}_A \in \mathrm{Rec}(\mathbb{R})$, where $A \subseteq \Sigma$. Moreover, let $\eta : \mathbb{M} \to S$ be a morphism recognizing $L$. Then it is easy to check that $\overrightarrow{L} = \bigcup_{(s,e)\in P} \mathbb{M}_s\mathbb{M}_e^\omega$ holds, with $P$ given by $P = \{ (s, e) \mid s \in \eta(L),\ se = s,\ e^2 = e \}$.
ii) The classical definition in the word case considers only infinite words. Here we have $L \subseteq \overrightarrow{L}$. According to our definition, we obtain the classical definition by the intersection $\overrightarrow{L} \cap \Sigma^\omega$. At least if one deals with traces our definition seems to

be more natural. In fact for word languages $L \subseteq \Sigma^{\infty}$ one usually intersects with $\Sigma^*$ and with $\Sigma^{\omega}$ and investigates the finitary and infinitary part separately. For traces, these intersections are replaced by intersections with $\mathbb{R}_A$ for $A \subseteq \Sigma$. It is also via these intersections how one can easily extend the result to complex traces.

The following technical proposition is an important tool for all our results. It gives a characterization of infinite traces which belong to the set $\overrightarrow{LK}$, for $L, K \subseteq \mathbb{M}$. The proof is deferred to the full version of the paper.

**Proposition 4.** *Given $(t_n)_{n\geq 0}, (w_n)_{n\geq 0} \subseteq \mathbb{M}$ such that $\{t_n w_n \mid n \geq 0\}$ is an infinite, directed set and let $x = \bigsqcup\{t_n w_n \mid n \geq 0\}$. Then there exist a subsequence of indices $(n_i)_{i\geq 0} \subseteq \mathbb{N}$ and sequences of finite traces $(s_i)_{i\geq 0}, (u_i)_{i\geq 0}, (v_i)_{i\geq 0} \subseteq \mathbb{M}$ such that $x = \bigsqcup\{t_{n_i} w_{n_i} \mid i \geq 0\}$ satisfying for $i \geq 0$ the following conditions:*

$$\begin{aligned} t_{n_i} &= s_0 u_0 \dots s_{i-1} u_{i-1} s_i, \\ w_{n_i} &= v_0 \dots v_{i-1} u_i v_i \\ \text{and} \quad & (v_i, s_j u_j) \in I, \quad \text{for } i < j. \end{aligned}$$

**Corollary 5.** *Given $L, K \subseteq \mathbb{M}$ with $K \in \mathrm{Rec}(\mathbb{M})$ such that $K\mathbb{M}_+ \cap K = \emptyset$. Let $(t_n)_{n\geq 0} \subseteq L$, $(w_n)_{n\geq 0} \subseteq K$ with $\{t_n w_n \mid n \geq 0\}$ be an infinite, directed set with $x = \bigsqcup\{t_n w_n \mid n \geq 0\}$. Let additionally $D(\mathrm{alphinf}(x)) = D(\mathrm{alph}(K))$.*
*Then there exist a subsequence of indices $(n_i)_{i\geq 0} \subseteq \mathbb{N}$ and sequences of finite traces $(s_i)_{i\geq 0}, (u_i)_{i\geq 0} \subseteq \mathbb{M}$ such that $x = \bigsqcup\{t_{n_i} w_{n_i} \mid i \geq 0\}$ and, for $i \geq 0$:*

$$t_{n_i} = s_0 u_0 \dots s_{i-1} u_{i-1} s_i \qquad \textit{and} \qquad w_{n_i} = u_i.$$

The next proposition generalizes the corresponding result for infinitary word languages (see also [PP91]) and will play a crucial role in the proof of the main result of this section. Apparently, the result differs from the analogous one for infinitary word languages, since we have an additional information about the alphabet at infinity. Nevertheless, observe that for $D = \Sigma \times \Sigma$, since $\mathbb{R}_e = \Sigma^{\omega}$ (if $e \neq 1$) we obtain indeed the result for $\Sigma^{\omega}$ as a special case of the next proposition (recall Rem. 1).

**Proposition 6.** *Let $S$ be a finite monoid, $\eta : \mathbb{M} \to S$ a morphism and $s, e \in S$ such that $se = s$ and $e \in E(S)$. Then we have the following inclusions:*

$$\mathbb{M}_s \mathbb{M}_e^{\omega} \quad \subseteq \quad \overrightarrow{\mathbb{M}_s \mathbb{P}_e} \cap \mathbb{R}_e \quad \subseteq \quad \bigcup_{f \leq e} \mathbb{M}_s \mathbb{M}_f^{\omega}$$

*Proof.* We omit the proof, which is analogous to the word case ([PP91]). We note that it is an easy consequence of Cor. 5, together with the principle of Ramsey factorizations.

**Corollary 7.** *Let $S$ be a finite monoid, $\eta : \mathbb{M} \to S$ a morphism and $s, e \in S$ such that $se = s$ and $e \in E(S)$. Then we have:*

1. $\mathbb{M}_e^\omega = \overrightarrow{\mathbb{M}_e\mathbb{P}_e} \cap \mathbb{R}_e$
2. $\bigcup_{f\leq e} \mathbb{M}_s\mathbb{M}_f^\omega \quad = \quad \bigcup_{f\leq e}(\overrightarrow{\mathbb{M}_s\mathbb{P}_f} \cap \mathbb{R}_f)$.

The following lemma expresses a property of the set $P_L = \{(s,e) \in S^2 \mid e \in E(S), se = s$ and $\mathbb{M}_s\mathbb{M}_e^\omega \cap L \neq \emptyset\}$ belonging to $L \in \mathrm{Rec}(\mathbb{R})$. Two pairs $(s,e),(s',e') \in S^2$ are called *conjugated* ([PP91]) if for some $x,y \in S$ the equalities $s' = sx$, $e = xy$ and $e' = yx$ hold.

**Lemma 8.** *Let* $L = \bigcup_{(s,e)\in P_L} \mathbb{M}_s\mathbb{M}_e^\omega$ *be recognized by a morphism* $\eta : \mathbb{M} \to S$, *with* $P_L$ *as above. Then for every elements* $s,s' \in S$, $e,e' \in E(S)$ *with* $(s,e),(s',e') \in P_L$, $\mathbb{M}_s\mathbb{M}_e^\omega \cap \mathbb{M}_{s'}\mathbb{M}_{e'}^\omega \neq \emptyset$ *if and only if* $(s,e)$ *and* $(s',e')$ *are conjugated.*

*Proof.* Similar to [PP91].

The next theorem is the main result of this section. It shows that every recognizable real trace language belongs to the Boolean closure of the family of deterministic languages. We obtain the well-known result for words as a special case of this theorem.

**Theorem 9.** *Let* $L \in \mathrm{Rec}(\mathbb{R})$ *be recognized by the morphism* $\eta : \mathbb{M} \to S$ *with* $S$ *a finite monoid and* $P_L = \{(s,e) \in S^2 \mid se = s,\ e \in E(S),\ \mathbb{M}_s\mathbb{M}_e^\omega \cap L \neq \emptyset\}$. *Then we have:*

$$L = \bigcup_{(s,e)\in P_L}\left(\bigcup_{f\leq e}(\overrightarrow{\mathbb{M}_s\mathbb{P}_f} \cap \mathbb{R}_f) \setminus \bigcup_{f<e}(\overrightarrow{\mathbb{M}_s\mathbb{P}_f} \cap \mathbb{R}_f)\right)$$

*Proof.* Consider $(s,e) \in P_L$ and $f\mathcal{R}e$ with $f \in E(S)$. Then, by Lemma 8, $(s,f) \in P_L$, since $(s,e)$ and $(s,f)$ are conjugated with $x = e$ and $y = f$ $(f = xy, e = yx, s = sx)$. Hence, $L = \bigcup_{(s,e)\in P_L} \mathbb{M}_s\mathbb{M}_e^\omega \subseteq \bigcup_{(s,e)\in P_L} \bigcup_{f\mathcal{R}e} \mathbb{M}_s\mathbb{M}_f^\omega \subseteq L$, where the first inclusion is due to $e\mathcal{R}e$. We obtain $L = \bigcup_{(s,e)\in P_L} \bigcup_{f\mathcal{R}e} \mathbb{M}_s\mathbb{M}_f^\omega$. Furthermore, we have $f\mathcal{R}e$ if and only if $ef = f$ and $fe = e$, hence if and only if $f \leq e$ and $f \not< e$.
Moreover, if $f < e$ holds, then $\mathbb{M}_s\mathbb{M}_e^\omega \cap \mathbb{M}_s\mathbb{M}_f^\omega = \emptyset$. $\quad (*)$
Otherwise, $(s,e)$ and $(s,f)$ would be conjugated (by Lemma 8) and one can easily check that this implies $e \leq f$, hence $f \not< e$.
We may now conclude the proof:

$$\bigcup_{f\mathcal{R}e} \mathbb{M}_s\mathbb{M}_f^\omega \;=\; \bigcup_{f\leq e, f\not< e} \mathbb{M}_s\mathbb{M}_f^\omega \;\overset{(*)}{=}\; \bigcup_{f\leq e} \mathbb{M}_s\mathbb{M}_f^\omega \setminus \bigcup_{f<e} \mathbb{M}_s\mathbb{M}_f^\omega \;=$$

$$= \bigcup_{f\leq e} \mathbb{M}_s\mathbb{M}_f^\omega \setminus \bigcup_{f<e}\bigcup_{g\leq f} \mathbb{M}_s\mathbb{M}_g^\omega \overset{\text{Cor.7}}{=}$$

$$\bigcup_{f\leq e}(\overrightarrow{\mathbb{M}_s\mathbb{P}_f} \cap \mathbb{R}_f) \setminus \bigcup_{f<e}\bigcup_{g\leq f}(\overrightarrow{\mathbb{M}_s\mathbb{P}_g} \cap \mathbb{R}_g) \;=$$

$$= \bigcup_{f\leq e}(\overrightarrow{\mathbb{M}_s\mathbb{P}_f} \cap \mathbb{R}_f) \setminus \bigcup_{f<e}(\overrightarrow{\mathbb{M}_s\mathbb{P}_f} \cap \mathbb{R}_f)$$

**Corollary 10.** $\mathrm{Rec}(\mathbb{R})$ *is equivalent to the Boolean closure of the family of deterministic real trace languages.*

## 4 Deterministic Asynchronous Automata for Rec($\mathbb{R}$)

In this section we will construct deterministic asynchronous cellular Muller automata for languages of the form $\overrightarrow{L}$ with $L \in \mathrm{Rec}(\mathbb{M})$. Since for every $\mathbb{R}_A$ ($A \subseteq \Sigma$) one can clearly exhibit a deterministic asynchronous cellular Muller automaton, we will obtain at the end the equivalence between Rec($\mathbb{R}$) and the family of languages accepted by deterministic asynchronous cellular Muller automata. Recall also from the definition of Muller automata that the accepted languages are closed under the Boolean operations.
We begin with a lemma which shows that we may restrict ourselves to a particular type of $\overrightarrow{L}$ with $L \in \mathrm{Rec}(\mathbb{M})$. Roughly speaking, we are interested in $\overrightarrow{L}$ where the alphabet at infinity $A$ is the same for all traces and where for every infinite trace in $\overrightarrow{L}$, its (infinitely many) $L$-prefixes have exactly one maximal element for each connected component of $A$.

**Lemma 11.** *Let $A = \bigcup_{i=1}^{k} A_i$ be a decomposition in connected components for $A \subseteq \Sigma$, i.e., $(A_i, A_j) \in I$ for $i \neq j$ and $A_i$ is connected for every $i = 1, \ldots, k$. Choose a fixed $a_i \in A_i$ for each $i$ and let $\mathbb{M}_{A,i}$ ($\mathbb{P}_{A,i}$ respectively) denote the following recognizable subsets of $\mathbb{M}$:*

$$\mathbb{M}_{A,i} = \{\, t \mid \mathrm{alph}(t) = A_i \text{ and } \max(t) = \{a_i\} \,\}$$
$$\mathbb{P}_{A,i} = \mathbb{M}_{A,i} \setminus \mathbb{M}_{A,i}\mathbb{M}_+$$

*Then we have for $L \subseteq \mathbb{M}$:*

$$\overrightarrow{L} \cap \mathrm{Inf}(A) = \overrightarrow{L\mathbb{P}_{A,1} \ldots \mathbb{P}_{A,k}} \cap \mathrm{Inf}(A).$$

*Proof.* The proof is similar to the proof of Proposition 6.

Before stating the main theorem of this section, let us recall some important concepts of Zielonka's construction of asynchronous cellular automata for Rec($\mathbb{M}$), which we will need in our construction (see also [Die90], [CMZ89]).
The $a$-prefix ($A$-prefix, respectively) $\partial_a(t)$ ($\partial_A(t)$, respectively) of a finite trace $t$ has been defined as the minimal prefix of $t$ containing all $a$ (all letters $a \in A$, respectively), which occur in $t$. More precisely, for every $a \in \Sigma$, $A \subseteq \Sigma$ and $t \in \mathbb{M}$,

$$\partial_a(t) = \sqcap\{\, u \leq t \mid |t|_a = |u|_a \,\}, \quad \partial_A(t) = \bigsqcup_{a \in A} \partial_a(t) \quad \text{(in particular } \partial_\emptyset(t) = 1\text{)}.$$

Zielonka's construction is based on the concept of *asynchronous mapping* (see e.g. [Die90], p. 49). It is a mapping $\varphi : \mathbb{M} \to Q$ to a set $Q$ satisfying the following 2 conditions, for every $t \in \mathbb{M}$, $a \in \Sigma$ and $A, B \subseteq \Sigma$:

- The value of $\varphi(\partial_{A \cup B}(t))$ is uniquely determined by $\varphi(\partial_A(t))$ and $\varphi(\partial_B(t))$.
- The value of $\varphi(\partial_{D(a)}(ta))$ is uniquely determined by $\varphi(\partial_{D(a)}(t))$ and by $a$.

Given an asynchronous mapping $\varphi : \mathbb{M} \to Q$ and a set $R \subseteq Q$, then an asynchronous cellular automaton $\mathcal{A} = (Q^{\Sigma}, \delta, q_0, F)$ with the following partial transition function accepts $\varphi^{-1}(R)$:

$$\begin{aligned}&\delta : Q^{\Sigma} \times \mathbb{M} \to Q^{\Sigma},\\ &\delta((\varphi(\partial_b(t)))_{b\in\Sigma}, a) = (\varphi(\partial_b(ta)))_{b\in\Sigma}\end{aligned}$$

(It is easy to check that $\delta$ is well-defined and satisfies the requirement for the partially defined transition function of an asynchronous cellular automaton, see e.g. [Die90], Prop. 2.4.4). The initial state of $\mathcal{A}$ is $q_0 = (\varphi(1), \dots, \varphi(1))$ and $F$ is given by $F = \{(\varphi(\partial_a(t)))_{a\in\Sigma} \mid \varphi(t) \in R\}$. Moreover, we have $\delta(q_0, t) = (\varphi(\partial_a(t)))_{a\in\Sigma}$.
Finally, suppose we are given a finitary recognizable language $K \subseteq \mathbb{M}$ recognized by a morphism $\eta : \mathbb{M} \to S$ onto a finite monoid $S$. Then, Zielonka's construction of an asynchronous cellular automaton accepting $K$ provides an asynchronous mapping $\varphi : \mathbb{M} \to Q$ to a finite set $Q$ and a mapping $\pi : Q \to S$ such that $\eta = \pi \circ \varphi$.
A crucial feature of the construction is the fact that the global state of the asynchronous cellular automaton based on $\varphi$ which is reached after having read a finite trace $t$, can be reconstructed by using the local states corresponding to the maximal elements of $t$: since $t = \partial_{\max(t)}(t)$ holds, we have $\varphi(t) = \varphi(\partial_{\max(t)}(t))$, which means that $\varphi(t)$ is exactly determined by $\{\varphi(\partial_a(t)) \mid a \in \max(t)\}$ (since $\varphi$ is an asynchronous mapping).

**Theorem 12.** *Let $L \in \mathrm{Rec}(\mathbb{M})$ be a recognizable finitary trace language. Then $\overrightarrow{L}$ can be recognized by a deterministic asynchronous cellular Muller automaton.*

*Proof.* Since we have

$$\overrightarrow{L} \;=\; \bigcup_{A\subseteq\Sigma}(\overrightarrow{L} \cap \mathrm{Inf}(A)) \;\overset{\text{Lemma 11}}{=}\; \bigcup_{A\subseteq\Sigma}(\overrightarrow{L\mathbb{P}_{A,1}\dots\mathbb{P}_{A,k}} \cap \mathrm{Inf}(A)),$$

with $\mathbb{P}_{A,1}, \dots \mathbb{P}_{A,k}$ depending on $A$ and defined as in the previous lemma, it will suffice to construct a deterministic Muller automaton accepting $\overrightarrow{L\mathbb{P}_{A,1}\dots\mathbb{P}_{A,k}} \cap \mathrm{Inf}(A)$.
Let $\eta : \mathbb{M} \to S$ be a morphism to a finite monoid recognizing $L\mathbb{P}_{A,1}\dots\mathbb{P}_{A,k}$, and let $\varphi : \mathbb{M} \to Q$ be the asynchronous mapping to the finite set $Q$ such that there exists a mapping $\pi$ with $\eta = \pi \circ \varphi$. Finally, consider the deterministic asynchronous cellular automaton $\mathcal{A}' = ((Q'_a)_{a\in\Sigma}, (\delta'_a)_{a\in\Sigma}, q'_0, F)$ accepting $L\mathbb{P}_{A,1}\dots\mathbb{P}_{A,k}$, which is obtained by Zielonka's construction. In particular, we have $\delta'(q'_0, t) = (\varphi(\partial_a(t)))_{a\in\Sigma}$, for every $t \in \mathbb{M}$, $a \in \Sigma$.
Furthermore, consider for every $f \in F$ the language

$$\begin{aligned}L_{A,f} = \{\, t \in \mathbb{R} \mid{}& \mathrm{alphinf}(t) = A,\ t = \bigsqcup\{t_n \mid n \geq 0\} \text{ with } t_0 \leq t_1 \leq \dots \text{ infinite},\\ & t_n \in L\mathbb{P}_{A,1}\dots\mathbb{P}_{A,k} \text{ and } \delta'(q'_0, t_n) = f, \text{ for every } n \geq 0 \,\}\end{aligned}$$

Clearly $\overrightarrow{L\mathbb{P}_{A,1}\dots\mathbb{P}_{A,k}} \cap \mathrm{Inf}(A) = \bigcup_{f\in F} L_{A,f}$ holds, since $\overrightarrow{K_1 \cup K_2} = \overrightarrow{K_1} \cup \overrightarrow{K_2}$, for $K_1, K_2 \subseteq \mathbb{M}$. Therefore, it suffices to construct a deterministic asynchronous cellular Muller automaton $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, \mathcal{T})$ accepting $L_{A,f}$. Let us define for $a \in \Sigma$:

$$Q_a = Q'_a \times \mathbb{Z}/2\mathbb{Z}$$
$$\delta_a((q,i)_{D(a)}) = (\delta'_a(q_{D(a)}), i_a + 1)$$
$$q_0 = (q'_{0a}, 0)_{a\in\Sigma}$$

Hence, we have $\delta_a(s_{D(a)}) \neq s_a$, for every $s_a \in Q_a$, $a \in \Sigma$. Thus, $|\mathrm{inf}_a(t)| \geq 2$ if and only if $a \in \mathrm{alphinf}(t)$, for each $a \in \Sigma$. We now define the table $\mathcal{T}$:

$$T = (T_a)_{a\in\Sigma} \in \mathcal{T} \quad \text{if and only if} \quad \begin{array}{l} \text{for some } i_a \in \mathbb{Z}/2\mathbb{Z} \\ T_a = \{(f_a, i_a)\} \text{ for } a \in \Sigma \setminus A, \\ (f_a, i_a) \in T_a \text{ and } |T_a| \geq 2, \text{ for } a \in A \end{array}$$

The inclusion $L_{A,f} \subseteq L(\mathcal{A})$ is not hard to be seen. Conversely, let $t \in L(\mathcal{A})$ be accepted by $T \in \mathcal{T}$. Clearly, $\mathrm{alphinf}(t) = \{a \in \Sigma \mid |T_a| \geq 2\} = A$, so let us factorize $t$ as $t = t_0 t_1 \ldots$ such that for some $i_a \in \mathbb{Z}/2\mathbb{Z}$, $a \in \Sigma$:

$$\begin{array}{ll} \mathrm{alph}(t_n) = A & \text{for } n \geq 1, \\ \delta(q_0, t_0 t_1 \ldots t_n)_a = (f_a, i_a) & \text{for } a \in (\Sigma \setminus A) \cup \{a_1, \ldots, a_k\} \text{ and } n \geq 0, \\ \max(t_0 \ldots t_n) \cap A = \{a_1, \ldots, a_k\} & \text{for } n \geq 0. \end{array}$$

(such a factorization exists because of the definition of the table $\mathcal{T}$ together with $a_i \in A_i$, with $(A_i, A_j) \in I$, for $i \neq j$).
We show that $t_0 t_1 \ldots t_n \in L\mathbb{P}_{A,1} \ldots \mathbb{P}_{A,k}$, $n \geq 0$. Evidently, $\delta'(q'_0, t_0 \ldots t_n)_a = f_a$, for $a \in (\Sigma \setminus A) \cup \{a_1, \ldots, a_k\}$, $n \geq 0$. From the definition of $\mathcal{A}'$, we know $f = (\varphi(\partial_a(u)))_{a\in\Sigma}$ for some $u \in L\mathbb{P}_{A,1} \ldots \mathbb{P}_{A,k}$. Note also that $\max(u) \cap A = \{a_1, \ldots, a_k\}$ and that for every $n \geq 0$, $a \in (\Sigma \setminus A) \cup \{a_1, \ldots, a_k\}$ the following holds:

$$(\delta'(q'_0, t_0 \ldots t_n)_a =) \quad \varphi(\partial_a(t_0 \ldots t_n)) = \varphi(\partial_a(u)) \quad (= f_a)$$

Since $\max(u), \max(t_0 \ldots t_n) \subseteq (\Sigma \setminus A) \cup \{a_1, \ldots a_k\}$, the observation made before we stated the theorem yields $\varphi(t_0 \ldots t_n) = \varphi(u)$, hence $\eta(t_0 \ldots t_n) = \eta(u)$, thus implying $t_0 \ldots t_n \in L\mathbb{P}_{A,1} \ldots \mathbb{P}_{A,k}$.

**Corollary 13.** *The family of recognizable real trace languages is equivalent to the family of languages, which are accepted by deterministic asynchronous cellular Muller automata.*

## 5 Deterministic $I$-diamond Muller automata

The rest of the paper can be read independently of the first part. It can also be omitted if the reader is interested only in the existence of deterministic asynchronous automata. We have included this section in this paper since due to the previous section we know for the first time that every closed recognizable language of $\Sigma^\omega$ can be recognized by some deterministic $I$-diamond Muller automaton. We give in the following no proofs, complexity results being deferred to a forthcoming paper.
Let $M$ be any monoid, then the classical definition [Eil74] defines a language $L \subseteq M$ to be recognizable if it is accepted by some deterministic $M$-automaton. In the

case of $M = \mathbb{M} = \mathbb{M}(\Sigma, D)$ an $\mathbb{M}$-automaton $\mathcal{A}$ is the same as a deterministic finite automaton $\mathcal{A} = (Q, \delta, q_0, F)$ over $\Sigma$ which satisfies the $I$-diamond property:

$$\forall q \in Q,\ (a,b) \in I : q \cdot ab = q \cdot ba$$

In some sense $I$-diamond finite automata do not accept traces directly, but they accept any representing word. So they accept closed languages.
Let us note that the $I$-diamond property does not ensure in the case of Muller automata that the accepted language is closed. We need some restriction of the tables in order to accept closed languages. We will give these restrictions for reduced tables, only. For this, consider a Muller automaton $\mathcal{A} = (Q, \delta, q_0, \mathcal{T})$. The table $\mathcal{T}$ is called reduced if for every $T \in \mathcal{T}$ there is some $u \in \Sigma^\omega$ such that $T$ represents the set of states occurring infinitely often on the path starting with $q_0$ and labelled by $u$. Let us denote the set of states on the path starting by a state $q \in Q$ and labelled by $v \in \Sigma^*$ by $\tau(q,v)$, i.e., $\tau(q,v) = \{\, qu \mid u \text{ is a prefix of } v \,\}$. The table $\mathcal{T}$ is called *closed* if for all $T \in \mathcal{T}, q \in T$ and $v \in \Sigma^*$ such that $qv = q$ and $T = \tau(q,v)$ we have $\tau(q,w) \in \mathcal{T}$, too, provided $w \in \Sigma^*$ denotes the same trace as $v$.
The next proposition is a special case of [GP91].

**Proposition 14 Gastin/Petit.** *Let $\mathcal{A} = (Q, \delta, q_0, \mathcal{T})$ be a deterministic $I$-diamond Muller automaton where $\mathcal{T}$ is reduced. Then $\mathcal{A}$ accepts a closed language if and only if the table $\mathcal{T}$ is closed.*

It is also stated in [GP91] that it is effectively decidable whether a reduced table is closed. However the decision procedure of [GP91] is based on the observation that the path length can be bounded. This does not yield any practical algorithm. However, giving an efficient decision procedure requires a subtler characterization of closed tables. Based on this characterization, we are able to show the NL completeness of the decision procedure. By NL we denote the class of problems decidable in non-deterministic logarithmic space, NSPACE (log(n)).

**Proposition 15.** *It is NL - complete to decide given a deterministic $I$-diamond Muller automaton $\mathcal{A} = (Q, \delta, q_0, \mathcal{T})$ over $(\Sigma, D)$ whether the table $\mathcal{T}$ is reduced.*

The proposition above shows that the preprocessing used to check the hypothesis of Proposition 14 is already NL-complete. However, the interesting fact is that even if we get this preprocessing for free, the test whether the accepted language is closed is still NL-complete.

**Proposition 16.** *There is an NL-algorithm to decide given a deterministic $I$-diamond Muller automaton $\mathcal{A} = (Q, \delta, q_0, \mathcal{T})$ over $(\Sigma, D)$, whether $L(\mathcal{A}) \subseteq \Sigma^\omega$ is closed.*

For the hardness we may even restrict us to a three letter alphabet.

**Theorem 17.** *Let $(\Sigma, D) = a - c - b$. It is NL-hard to decide given a deterministic $I$-diamond Muller automaton $\mathcal{A} = (Q, \delta, q_0, \mathcal{T})$ where $\mathcal{T}$ is reduced whether $L(\mathcal{A})$ is closed.*

## 6 Conclusion

In this paper we gave a characterization of recognizable real trace languages by deterministic asynchronous Muller automata, thus answering one of the main open problems about infinite traces, see e.g. [GP92]. We showed that classical results of the theory of recognizable infinitary word languages have a natural extension in the case of real traces. One open problem which arises is whether there exists a characterization of $\overrightarrow{L}$ with $L \in \mathrm{Rec}(\mathbb{M})$ (i.e., without intersecting with sets $\mathbb{R}_A$) by means of deterministic asynchronous Büchi automata.

**Acknowledgements:** We thank Paul Gastin for fruitful discussions and Wieslaw Zielonka who told us that independently he has developed similar ideas. We also thank the anonymous referees from STACS for many valuable comments.

## References

[CMZ89] R. Cori, Y. Métivier, and W. Zielonka. Asynchronous mappings and asynchronous cellular automata. Tech. rep. 89-97, LABRI, Univ. Bordeaux, 1989.

[Die90] V. Diekert. *Combinatorics on Traces.* Number 454 in Lecture Notes in Computer Science. Springer, Berlin-Heidelberg-New York, 1990.

[Eil74] S. Eilenberg. *Automata, Languages and Machines*, volume I. Academic Press, New York and London, 1974.

[Gas91] P. Gastin. Recognizable and rational trace languages of finite and infinite traces. In Choffrut C. et al., editors, *Proceedings of the 8th Annual Symposium on Theoretical Aspects of Computer Science (STACS'91), Hamburg 1991*, number 480 in Lecture Notes in Computer Science, pages 89–104. Springer, Berlin-Heidelberg-New York, 1991.

[GP91] P. Gastin and A. Petit. Büchi (asynchronous) automata for infinite traces. Tech. rep., LRI, Université Paris-Sud, 1991.

[GP92] P. Gastin and A. Petit. Asynchronous cellular automata for infinite traces. In W. Kuich, editor, *Proceedings of the 19th International Colloquium on Automata Languages and Programming (ICALP'92), Vienna (Austria) 1992*, Lecture Notes in Computer Science. Springer, Berlin-Heidelberg-New York, 1992. Also available as Tech. Rep. 91-68, LITP, Université Paris 6, France, 1991.

[McN66] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9:521–530, 1966.

[PP91] D. Perrin and J.E. Pin. Mots Infinis. Technical Report, LITP 91.06, Université de Paris 6, 1991. Book to appear.

[Tho90] W. Thomas. Automata on infinite objects. In Jan van Leeuwen, editor, *Handbook of theoretical computer science*, pages 133–191. Elsevier Science Publishers, 1990.

[Zie87] W. Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O.-Informatique Théorique et Applications*, 21:99–135, 1987.

[Zie89] W. Zielonka. Safe executions of recognizable trace languages by asynchronous automata. In A. R. Mayer et al., editors, *Proceedings Symposium on Logical Foundations of Computer Science, Logic at Botik '89, Pereslavl-Zalessky (USSR) 1989*, number 363 in Lecture Notes in Computer Science, pages 278–289. Springer, Berlin-Heidelberg-New York, 1989.