Deterministic asynchronous automata for infinite traces*

Volker Diekert Anca Muscholl

Universität Stuttgart
Institut für Informatik
Breitwiesenstr. 20-22
D-70565 Stuttgart

Abstract

This paper shows the equivalence between the family of recognizable languages over infinite traces and the family of languages which are recognized by 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.

1 Introduction

A. Mazurkiewicz introduced the concept of traces as a suitable semantics for concurrent systems [Maz77]. A concurrent system is given by a set of atomic actions $\Sigma = \{a, b, c, \ldots\}$ together with an independence relation $I \subseteq \Sigma \times \Sigma$, which specifies pairs of actions which can be performed concurrently. This leads to an equivalence relation on Σ^* generated by the independence relation I. More precisely, if a and b denote independent actions, then for $u, v \in \Sigma^*$, $uabv$ and $ubav$ are equivalent. They can be interpreted as two sequential observations of the same concurrent run. Since this equivalence relation turns out to be a congruence, it defines a quotient monoid of Σ^*, which is called trace monoid and is denoted by $\mathbb{M}(\Sigma, D)$. In fact, these monoids have been introduced and studied independently by Cartier and Foata [CF69] in combinatorics and are also called free partially commutative monoids.

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

Hence, a trace is a congruence class of words. From a different point of view, a trace is a finite labelled acyclic graph, where the labels of vertices are actions and edges represent the ordering between dependent actions. Such a graph is called dependence graph and the semantics is that every execution of the concurrent run has to respect the induced partial order.

In order to describe non-sequential processes which never terminate, e.g. distributed operating systems, the notion of infinite trace was needed. One of the first explicit definition of infinite traces was given again by Mazurkiewicz [Maz87]. He defines an (infinite) trace as an (infinite) prefix closed directed subset of finite traces. This is the same as an (infinite) dependence graph where each vertex has finitely many predecessors only [Maz87, Thm.13]. In the following these objects are called here *real traces*. Unfortunately, there can be no convenient associative concatenation on real traces. This led to the definition of the monoid of complex traces [Die91]. It is a quotient monoid of the set of infinite dependence graphs under the largest congruence which respects real parts.

Recognizable subsets form a well-studied family in the context of both free partially commutative monoids and (in)finite sequences. They may be defined by means of saturating morphisms to a finite monoid or, more commonly, by means of recognizability with finite automata. A further approach is based on rational (c-rational respectively) expressions, whereby equivalent characterizations are given by Kleene's and Büchi's theorems for word languages (Ochmanski's theorem for trace languages, respectively). Finally, recognizability is equivalent to definability in certain monadic second order theories.

As regards recognizable real trace languages, there exist equivalent characterizations by means of recognizing morphisms [Gas91] and by c-rational expressions [GPZ91]. Concerning characterizations by finite automata, P. Gastin and A. Petit investigated asynchronous (cellular) automata for infinite traces [GP92]. This type of automaton has been defined by W. Zielonka [Zie87, Zie89], who showed that for finite traces, the recognizable languages are exactly the languages recognized by deterministic asynchronous (cellular) automata. Asynchronous (cellular) automata have a decentralized control, which allows the parallel execution of independent actions. This concurrent behaviour makes them appropriate for the recognition of trace languages. For infinite traces, adequate acceptance conditions were needed. Gastin/Petit used a suitable Büchi condition which led to a characterization of recognizable real trace languages by nondeterministic asynchronous (cellular) Büchi automata. The major open problem which arose and which we are going to solve here was a characterization by some kind of deterministic automata.

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. (In the theory of recognizable word languages the theorem of McNaughton states the equivalence between nondeterministic Büchi automata and deterministic Muller automata.) 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. 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 basic notions. We also recall some facts about recognizable infinitary word and trace languages. Section 3 gives first a technical proposition which allows to follow the approach of Perrin/Pin [PP92][1] to McNaughton's Theorem. We show as the main result of this section that the family of recognizable real trace languages is equivalent to the Boolean closure of a family of languages, which in the word case coincides with the family of deterministic word 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. As asynchronous automata are a special form of so-called I-diamond automata, we directly obtain that recognizable, closed (w.r.t. the independence relation I) infinitary word languages can be accepted by deterministic I-diamond Muller automata. I-diamond automata are a much weaker model for concurrency. They have global states only and concurrency is expressed by interleaving, i.e., for a global state q and a pair (a, b) of independent actions we have $q \cdot ab = q \cdot ba$. Although this is a weaker model, no direct proof for the existence of deterministic I-diamond automata is known up to now.

2 Preliminaries

2.1 Basic Notions

We denote by (Σ, D) a finite *dependence alphabet*, with Σ 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 $D(a) = \{b \in \Sigma \,|\, (a, b) \in D\}$ and $D(A) = \bigcup_{a \in A} D(a)$ for $A \subseteq \Sigma$.

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 are congruence classes of words, and they can be identified with their dependence graph, i.e. with (isomorphism classes of) labelled, acyclic, directed graphs $[V, E, \lambda]$, where V is any set of vertices labelled by $\lambda : V \to \Sigma$ and E is a set of edges with the property that $E \cup E^{-1} \cup \mathrm{id}_V = \lambda^{-1}(D)$. Given a trace $t = [a_1 \cdots a_n] \in \mathbb{M}$, $a_i \in \Sigma$ for $1 \le i \le n$, we obtain the dependence graph of t by taking first a set of n vertices $V = \{v_1, \ldots, v_n\}$. Then we label node v_i with a_i, i.e. $\lambda(v_i) = a_i$, for $1 \le i \le n$ and we define the edge set by

[1]This work is not broadly available yet. Nevertheless, we give all relevant proofs directly.

$E = \{(v_i, v_j) \mid i < j$ and $(\lambda(v_i), \lambda(v_j)) \in D\}$. The notion of dependence graph 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 us to represent vertices as pairs (a, i), with $a \in \Sigma$ and i a countable ordinal, where (a, i) represents the $(i+1)$-th node labelled by the letter a. This is called the *standard representation.*

$\mathbb{G}(\Sigma, D)$ 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 ω-iteration as $L^\omega = \{\, g_0 g_1 \ldots \mid g_n \in L,\ \forall n \geq 0 \,\}$.

We denote by Σ^ω the set of infinite words over the alphabet Σ (i.e. mappings from $\mathbb{N}$ to Σ), and let $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. Languages $L \subseteq \Sigma^\infty$ are called *infinitary* word languages (whereas subsets of Σ^* are called *finitary* languages).

The canonical homomorphism $\varphi : \Sigma^* \to \mathbb{M}(\Sigma, D)$ can be extended to a mapping $\varphi : \Sigma^\infty \to \mathbb{G}(\Sigma, D)$. 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 for example, $a^\omega \in \mathbb{R}(\Sigma, D)$, $a \in \Sigma$, but the concatenation in $\mathbb{G}(\Sigma, D)$ yields a nonreal graph, where one node is transfinite. In fact, φ commutes neither with concatenation nor with ω-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 (Σ, 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. For $t \in \mathbb{R}$ we denote by $\mathrm{alph}(t)$ the set of letters occurring in t, and by $|t|_a \in \mathbb{N} \cup \{\omega\}$, $a \in \Sigma$, the number of a occurring in t. By $\mathrm{alphinf}(t)$ we denote the set of letters occurring infinitely often in t, i.e., $\mathrm{alphinf}(t) = \{a \in \Sigma \mid |t|_a = \omega\}$. 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 $t = us$. For traces t_1, t_2 we denote by $t_1 \sqcap t_2$ their greatest lower bound. Whenever it exists, the least upper bound of a set $Y \subseteq \mathbb{R}$ w.r.t. the prefix ordering is denoted by $\sqcup Y$.

The best way to understand our formal proofs is perhaps the graphical interpretation of traces. As an example let us sketch the proof for every directed set of real

traces having a least upper bound [GR91, Kwi90]. Consider each trace t in its standard representation with $V = \{(a, i) \mid 0 \leq i < |t|_a\}$ and $E = \{((a, i), (b, j)) \mid$ the $(i+1)^{\text{th}}$ a is before the $(j+1)^{\text{th}}$ b in any representing word of $t\}$. Then, for two traces $t_1 = [V_1, E_1, \lambda_1]$, $t_2 = [V_2, E_2, \lambda_2]$ we may define $t_1 \cup t_2$ by the set theoretical union of the dependence graphs. In general, $t_1 \cup t_2$ will be no trace. However, this is the case if $t_1 \leq t$ and $t_2 \leq t$ for some t. Indeed, in this case there will be no cycles in the resulting graph. More general, if $X \subseteq \mathbb{R}$ is directed, then the union $\bigcup_{t \in X} t$ is a well-defined dependence graph of some real trace and we obtain $\sqcup X = \bigcup_{t \in X} t$.
Another useful application of the dependence graph view is the identification of factors. Assume that we can write $t = u_1 u_2 u_3$ for some $u_1, u_2 \in \mathbb{M}$ and $u_3 \in \mathbb{R}$. Then we can identify the factors u_1, u_2 and u_3 with subgraphs of t (called subtraces). This generalizes in a natural way to more factors. If we have $t = u_1 \cdots u_m = v_1 \cdots v_n$, then we can define factors w_{ij} for $1 \leq i \leq m$ and $1 \leq j \leq n$ by the intersections $u_i \cap v_j$ in the dependence graph of t. We obtain $u_i = w_{i1} \cdots w_{in}$ and $v_j = w_{1j} \cdots w_{mj}$. Furthermore, an inspection of the dependence graph of t shows that there are no edges from w_{ij} to w_{kl} whenever $i < k$ and $j > l$ ($i > k$ and $j < l$, resp.). Hence, $\text{alph}(w_{ij}) \times \text{alph}(w_{kl}) \subseteq I$ in this cases. This is the main idea behind various proofs given below. For the formal reasonings however, it is in most case sufficient to consider the special case $m = n = 2$, which is exactly the well-known Levi's Lemma, see Lemma 3.5 below.

2.2 Recognizable word languages

The family of *recognizable* infinitary word languages, denoted by $\text{Rec}(\Sigma^\infty)$, can be defined by means of recognizing morphisms or equivalently, by finite state automata with suitable acceptance conditions for infinite words (see [Tho90]).
Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F_0)$ 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_0 is the set of accepting states. Let $\pi = (q_0, a_0, q_1, a_1, \ldots)$ with $q_n \in Q$, $a_n \in \Sigma$ and $q_{n+1} \in \delta(q_n, a_n)$, denote an infinite transition path in $\mathcal{A}$. Then $\inf(\pi)$ is the set of states occurring infinitely often in π. A Büchi automaton is a finite automaton augmented by a state set $F \subseteq Q$, i.e. $\mathcal{A} = (Q, \Sigma, \delta, q_0, F_0, F)$. Having two types of final states $\mathcal{A}$ recognizes infinitary languages as follows. Finite words are accepted by reaching a final state in F_0. An infinite word w is accepted by the Büchi automaton $\mathcal{A}$ if there exists a transition path π of $\mathcal{A}$ labelled by w, i.e. $w = a_0 a_1 \ldots$, such that $f \in \inf(\pi)$ for some final state $f \in F$. For the Muller acceptance condition, we have a state table $\mathcal{T} \subseteq \mathcal{P}(Q)$ instead of the state set F. An infinite word w is accepted by the Muller automaton $\mathcal{A} = (Q, \Sigma, \delta, q_0, F_0, \mathcal{T})$ if there is a path π labelled by w such that $\inf(\pi) = T$ for some $T \in \mathcal{T}$. Finite words are accepted by $\mathcal{A}$ as above, by reaching a final state of F_0.
A language $L \subseteq \Sigma^\infty$ is called *recognizable* if it is accepted by some nondeter-

ministic Büchi automaton. Furthermore, deterministic Büchi automata do not suffice for characterizing $\mathrm{Rec}(\Sigma^\infty)$. However, the well-known theorem of McNaughton [McN66] shows that $\mathrm{Rec}(\Sigma^\infty)$ coincides with the family of infinitary word languages accepted by deterministic Muller automata (see e.g. [Tho90]).
Finally, let us mention that for infinitary word languages Büchi's theorem holds, i.e., the family of recognizable languages is identical with the family defined by rational expressions over Σ using the operations union, concatenation, $*$-iteration, and ω-iteration.

2.3 Recognizable Real Trace Languages

One possible way to define the family of *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 monoid S. A real trace language $L \subseteq \mathbb{R}$ is *recognized* by η if for any sequence of finite traces $(t_n)_{n\geq 0} \subseteq \mathbb{M}$ the following saturation condition holds:

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

An equivalent definition uses the syntactic congruence of Arnold ([Arn85]). For $L \subseteq \mathbb{R}$, two finite traces $u, v \in \mathbb{M}$ are syntactically congruent if and only if :

$$\forall x, y \in \mathbb{M} : \quad x(uy)^\omega \in L \;\Leftrightarrow\; x(vy)^\omega \in L$$

$$\forall x, y, z \in \mathbb{M} : \quad xuyz^\omega \in L \;\Leftrightarrow\; xvyz^\omega \in L$$

We denote the syntactic congruence by $\equiv_L$ and consider the canonical morphism $\eta : \mathbb{M} \to \mathrm{Synt}(L)$, where $\mathrm{Synt}(L) = \mathbb{M}/\!\equiv_L$ is the syntactic monoid of L.
The following definition summarizes some of the equivalent characterizations of $\mathrm{Rec}(\mathbb{R})$ (see [Gas91], [GP92]).

Definition 2.1 *A real trace language L is* recognizable, *$L \in \mathrm{Rec}(\mathbb{R})$, if one of the following equivalent conditions holds:*

1. *There exists a morphism $\eta : \mathbb{M} \to S$ to a finite monoid S recognizing L.*

   (Moreover, L can be represented as $L = \bigcup_{(s,e)\in P} \eta^{-1}(s)\, \eta^{-1}(e)^\omega$ with

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

2. *$\varphi^{-1}(L) \subseteq \Sigma^\infty$ is a recognizable word language, i.e. $\varphi^{-1}(L) \in \mathrm{Rec}(\Sigma^\infty)$.*

3. *The syntactic congruence $\equiv_L$ has finite index and $\eta : \mathbb{M} \to Synt(L)$ recognizes L.*

   (Moreover, we have $Synt(L) = Synt(\varphi^{-1}(L))$.)

4. *L is accepted by a nondeterministic Büchi asynchronous cellular automaton (see the definition below).*

Remark 2.2 For the first condition above note that if $\eta : \mathbb{M} \to S$ recognizes a language L, then we have for any $s, e \in S$: $(\eta^{-1}(s)\,\eta^{-1}(e)^\omega \cap L \neq \emptyset) \implies (\eta^{-1}(s)\,\eta^{-1}(e)^\omega \subseteq L)$.

Languages $L \subseteq \mathbb{M}$ of finite traces are called finitary in the following. For the family of finitary recognizable trace languages $\mathrm{Rec}(\mathbb{M})$, W. Zielonka introduced the concept of asynchronous (cellular) automata ([Zie87], [Zie89]) and showed the deep result of the equivalence between $\mathrm{Rec}(\mathbb{M})$ and the family of finitary trace languages accepted by deterministic asynchronous (cellular) automata.
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 local 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:

$$q' \in \delta(q,a) \;\Leftrightarrow\; \begin{array}{ll} q'_a \in \delta_a(q_{D(a)}) & \text{and} \\ q'_c = q_c, & \text{for all } c \neq a \end{array}$$

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 (Concurrent Read Owner Write, CROW).
In [GP92] a suitable counterpart of the Büchi and Muller acceptance conditions has been defined for infinite traces. Consider an infinite transition path $\pi = (q_0, a_0, q_1, a_1, \ldots)$ in $\mathcal{A}$, with $q_n \in Q_\Sigma$, $a_n \in \Sigma$ and $q_{n+1} \in \delta(q_n, a_n)$ for $n \geq 0$. The acceptance conditions will concern the sets of local states which occur infinitely often in π. Consider for each $a \in \Sigma$ the set of local a-states $\mathrm{inf}_a(\pi) = \{\, q_a \in Q_a \mid q_{n,a} = q_a \text{ for infinitely many } n \,\}$. If $\mathcal{A}$ is deterministic, then we will write directly $\mathrm{inf}_a(t)$ instead of $\mathrm{inf}_a(\pi)$, where $t = \varphi(a_0 a_1 \ldots)$.
In order to accept infinite traces, an asynchronous cellular automaton is augmented by 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 $\mathcal{A}$ with a *Büchi* condition if there exist an infinite transition path $\pi = (q_0, a_0, q_1, a_1, \ldots)$ labelled by some representing word of t (i.e. $t = \varphi(a_0 a_1 \ldots)$ holds) and a set $T \in \mathcal{T}$ such that $\mathrm{inf}_a(\pi) \supseteq T_a$ for every $a \in \Sigma$.
The automaton $\mathcal{A}$ accepts $t \in \mathbb{R}$ with a *Muller* condition if there exist an infinite path π and a set $T \in \mathcal{T}$ as above with $\mathrm{inf}_a(\pi) = T_a$ for every $a \in \Sigma$. Thus, the difference is that the Muller condition specifies exactly the set of (local) states which have to be repeated infinitely often.

Finite traces are accepted in the usual way, by reaching a final state.
Let us illustrate the acceptance mode by a very simple example:

Example 2.3 Let $b, c \in \Sigma$, $b \neq c$ and consider $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, F, \mathcal{T})$ with

1. $Q_a = \{q_{0,a}\}$ and $\delta_a(q_{D(a)}) = q_{0,a}$ for every $q \in Q_\Sigma$ and $a \neq b, c$.
2. For $x \in \{b, c\}$ let $Q_x = \{q_{0,x}, q_{1,x}\}$ and
$$\delta_x(q_{D(x)}) = \begin{cases} q_{1,x} & \text{if } q_x = q_{0,x} \\ q_{0,x} & \text{if } q_x = q_{1,x} \end{cases}$$
3. $F = \emptyset$.
4. $\mathcal{T} = \{T\}$ with $T = (\prod_{a\in\Sigma\setminus\{b,c\}} Q_a) \times Q_b \times \{q_{0,c}\}$.

Clearly we have using the Muller acceptance condition: $L(\mathcal{A}) = \{t \in \mathbb{R} \mid b \in \text{alphinf}(t), |t|_c \text{ finite, even}\}$, whereas using the Büchi condition, $L(\mathcal{A}) = \{t \in \mathbb{R} \mid b \in \text{alphinf}(t), \text{ and } (|t|_c = \omega \text{ or } |t|_c \text{ even})\}$.
Note also that the family of languages accepted by deterministic asynchronous cellular Muller automata is closed under Boolean operations. Moreover, for every $A \subseteq \Sigma$ we can exhibit deterministic asynchronous cellular Muller automata recognizing the sets $\text{Inf}(A) := \{t \in \mathbb{R} \mid \text{alphinf}(t) = A\}$ resp. $\mathbb{R}_A := \{t \in \mathbb{R} \mid D(\text{alphinf}(t)) = D(A)\} = \bigcup_{B:\,D(B)=D(A)} \text{Inf}(B)$. It is enough to modify the automaton above correspondingly.

Note that the acceptance conditions are inherently local, thus being suitable for automata with decentralized control. Moreover, the word language accepted by asynchronous cellular automaton with the above acceptance conditions is closed [GP92], a property which we implicitly used in the definition of the acceptance of a trace. Gastin/Petit showed the equivalence between Rec($\mathbb{R}$) and the family of real trace languages which are accepted by nondeterministic Büchi asynchronous cellular automata (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 construction (even considering different acceptance conditions) in order to obtain a deterministic automaton. The construction below is based on a totally different approach.
In this paper we consider only asynchronous cellular Muller automata, due to their elementary internal structure, together with a simple construction allowing the transformation of asynchronous cellular Muller automata into equivalent asynchronous Muller automata. Therefore our results will hold for asynchronous Muller automata as well.

3 Algebraic Results on Recognizable Real Trace Languages

In this section we consider a trace language $L \in \mathrm{Rec}(\mathbb{R})$ which is recognized by a homomorphism $\eta : \mathbb{M} \to S$ to a finite monoid S and we use the following notations:

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

Thus, $\mathbb{M}_s$ is the set of all finite traces mapped by η to $s \in S$ and $\mathbb{P}_s$ is the subset of $\mathbb{M}_s$ consisting of traces having no proper prefix in $\mathbb{M}_s$.
Throughout the whole section we will assume that η is surjective and that we have $\mathrm{alph}(t) = \mathrm{alph}(t')$ for all $t, t' \in \mathbb{M}$ with $\eta(t) = \eta(t')$. This can be easily achieved, since we may replace S by a submonoid of $S \times \mathcal{P}(\Sigma)$, with the multiplication defined by $(s, A)(s', A') = (ss', A \cup A')$ and $(1, \emptyset)$ as identity and where the morphism is replaced by $t \mapsto (\eta(t), \mathrm{alph}(t))$. Moreover, we define $\mathrm{alph}(s)$ for $s \in S$ by $\mathrm{alph}(s) = \mathrm{alph}(t)$ for some $t \in \eta^{-1}(s)$.
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 subset of S of idempotent elements, $E(S) = \{ e \in S \mid e = e^2 \}$. For $s \in S$ define the partial order relation $\leq$ on the set $\{e \in E(S) \mid se = s\}$ by $f \leq e$ if and only if $ef = f$ (actually, $f \leq e$ is equivalent to $f \leq_{\mathcal{R}} e$). By $f < e$ we mean $f \leq e$ and $e \not\leq f$. In fact, everytime we will use languages of the type $\mathbb{M}_s\mathbb{M}_e^\omega$ or $\mathbb{M}_s\mathbb{P}_e$ we will suppose implicitly that the relations $se = s$ and $e \in E(S)$ hold.
Back to infinite traces, recall the notations from the previous section

$$\begin{array}{l} \mathrm{Inf}(A) = \{ t \in \mathbb{R} \mid \mathrm{alphinf}(t) = A \}, \\ \mathbb{R}_A = \{ t \in \mathbb{R} \mid D(\mathrm{alphinf}(t)) = D(A) \} \quad \text{for } A \subseteq \Sigma, \end{array}$$

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

Remark 3.1 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 $\mathbb{R}_\emptyset = \Sigma^*$. Therefore we prefer to use $\mathbb{R}_s$ instead of $\mathrm{Inf}(s)$, since so we obtain the analogous results for infinite words as a special case of our results.

Definition 3.2 *A nonempty set $Y \subseteq \mathbb{M}$ is called directed if for every $t, t' \in Y$, there exists an upper bound of t, t' which also belongs to Y.*

As shown in [GR91, Kwi90] every directed set of real traces admits a least upper bound, see also Section 2.1.

Definition 3.3 *Let $L \subseteq \mathbb{M}$.*
We define $\overrightarrow{L} := \{ t \in \mathbb{R} \mid t = \sqcup Y$ for some directed $Y \subseteq L \}$.

Remark 3.4 i) The classical definition of $\overrightarrow{L}$ in the word case considers only infinite words. It corresponds to the intersection $\overrightarrow{L} \cap \Sigma^\omega$ in our definition. Here we have $L \subseteq \overrightarrow{L}$. 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 Σ^* and with Σ^ω and investigates the two parts separately. For traces, these intersections will be replaced by intersections with $\mathbb{R}_A$ for $A \subseteq \Sigma$. It is also via these intersections how one can easily extend the results to complex traces.
ii) Note that every $\overrightarrow{L}$ with $L \in \mathrm{Rec}(\mathbb{M})$ is recognizable. 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 $P = \{ (s,e) \mid s \in \eta(L),\ se = s,\ e^2 = e \}$.

The following technical propositions represent important tools for all our results. We begin by recalling the well-known decomposition lemma (Levi's Lemma) for traces:

Lemma 3.5 ([CP85]) *Let $t_1, t_2, x_1, x_2 \in \mathbb{M}$ satisfying $t_1x_1 = t_2x_2$. Then*

$$t_1 = pf,\ t_2 = pg,\ x_1 = gy,\ \textit{and}\ x_2 = fy$$

for $p = t_1 \sqcap t_2$ and some $f, g, y \in \mathbb{M}$ such that $\mathrm{alph}(f) \times \mathrm{alph}(g) \subseteq I$.

Lemma 3.6 *Let $(t_n)_{n\geq 0} \subseteq \mathbb{M}$ be a sequence admitting a least upper bound. Then there exists an infinite index set $J \subseteq \mathbb{N}$ such that $(t_i)_{i\in J}$ is an increasing sequence (w.r.t. the prefix order).*

Proof. Since $\bigsqcup_{n\geq 0} t_n$ exists, let us consider for $i < j$ the greatest lower bounds $p_{i,j} := t_i \sqcap t_j$ and traces $f_{i,j}, g_{i,j}$ such that according to Lemma 3.5 the following holds:

$$t_i = p_{i,j}\, f_{i,j},\ t_j = p_{i,j}\, g_{i,j} \text{ and } \mathrm{alph}(f_{i,j}) \times \mathrm{alph}(g_{i,j}) \subseteq I$$

Fixing i clearly allows us to assume $p_{i,j} = p_i$ (resp. $f_{i,j} = f_i$) for some p_i (resp. f_i) and every $j > i$. (This can be achieved by taking an appropriate (infinite) subsequence of indices.) Hence we may rewrite the above equations:

$$t_i = p_i f_i,\ t_j = p_i\, g_{i,j} \text{ and } \mathrm{alph}(f_i) \times \mathrm{alph}(g_{i,j}) \subseteq I$$

Observe now that $p_i \leq p_j$ for every $i \leq j$, since by construction, $p_i \leq t_j$ and $p_i \leq t_k$, for every $k \geq j$. Thus, $p_i \leq p_j\ (= t_j \sqcap t_k$ for every $k \geq j)$. Furthermore we have for every $i < j$:

$$t_j = p_i\, g_{i,j} = p_j f_j.$$

Together with $\mathrm{alph}(f_j) \subseteq \mathrm{alph}(g_{i,j})$ (since $p_i \leq p_j$), this yields $\mathrm{alph}(f_i) \times \mathrm{alph}(f_j) \subseteq I$. Since Σ is a finite alphabet, this implies that $f_i = 1$ for all but finitely many i. By a further restriction of the index set such that $f_i = 1$ for every $i \in J$ we obtain an (infinite) increasing sequence $(t_i)_{i\in J}$. $\square$

Remark 3.7 Note that the lemma above does not claim $\bigsqcup_{i\in J} t_i = \bigsqcup_{n\geq 0} t_n$. It is easy to find counterexamples. Consider $\Sigma = \{a,b,c\}$, $D = \{(a,a),(b,b),(c,c)\}$ and

$$t_n = \begin{cases} a^2bc^n & \text{for even } n \\ ab^2c^n & \text{for odd } n \end{cases}$$

Now one has $\bigsqcup_{n\geq 0} t_n = a^2b^2c^\omega$, but neither $\bigsqcup_{n\geq 0} t_{2n} = a^2bc^\omega$ nor $\bigsqcup_{n\geq 0} t_{2n+1} = ab^2c^\omega$ is equal to $a^2b^2c^\omega$.

Proposition 3.8 *Given $(t_n)_{n\geq 0}, (w_n)_{n\geq 0} \subseteq \mathbb{M}$ such that $\{\, t_nw_n \mid n \geq 0 \,\}$ is an infinite, directed set and let $x = \bigsqcup\{\, t_nw_n \mid n \geq 0 \,\}$. Then there exist a subsequence of indices $(n_i)_{i\geq 0}$ 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, j the following conditions:*

$$\begin{aligned} t_{n_i} &= s_0u_0\cdots s_{i-1}u_{i-1}s_i, \\ w_{n_i} &= u_iv_0\cdots v_{i-1}v_i \\ \textit{and} \quad & \mathrm{alph}(v_i)\times \mathrm{alph}(s_ju_j) \subseteq I, \quad \textit{for } i<j. \end{aligned}$$

Proof. Since the set $\{\, t_nw_n \mid n \geq 0 \,\}$ is directed, we may assume by switching to an appropriate infinite subsequence of indices that $t_nw_n \leq t_{n+1}w_{n+1}$ for all $n \geq 0$. Moreover, applying Lemma 3.6 allows us to consider $(t_n)_{n\geq 0}$ as an increasing sequence, too. Therefore, for $0 \leq i < j$ we can define $(x_{i,j})_{i,j\geq 0} \subseteq \mathbb{M}$ such that $t_j = t_i\, x_{i,j}$.
By the left-cancellation property of $\mathbb{M}$ the above relations yield $w_i \leq x_{i,j}\, w_j$. Applying now Levi's Lemma to this inequation we obtain traces $u_{i,j}, s_{i,j}, y_{i,j}$ such that $\mathrm{alph}(s_{i,j}) \times \mathrm{alph}(y_{i,j}) \subseteq I$ and

$$w_i = u_{i,j}\, y_{i,j},\ x_{i,j} = u_{i,j}\, s_{i,j},\ y_{i,j} \leq w_j.$$

As we are interested in an (infinite) subsequence of indices, for a fixed $i \geq 0$ we may assume $u_{i,j} = u_i$ and $y_{i,j} = y_i$ for some $u_i, y_i \in \mathbb{M}$ and for every $i < j$. Hence, we may rewrite the above relations as $\mathrm{alph}(s_{i,j}) \times \mathrm{alph}(y_i) \subseteq I$ and

$$w_i = u_iy_i,\ x_{i,j} = u_i\, s_{i,j},\ y_i \leq w_j.$$

Let us denote $s_{i,i+1}$ by s_{i+1} and $x_{i,i+1}$ by x_i, then we obtain using $x_{i,j} = u_i\, s_{i,j}$,

$$x_{i,j} = x_ix_{i+1}\cdots x_{j-1} = (u_is_{i+1})(u_{i+1}s_{i+2})\cdots(u_{j-1}s_j) = u_i\, s_{i,j},$$

hence, $s_{i,j} = s_{i+1}u_{i+1}s_{i+2}\cdots u_{j-1}s_j$. Together with the above alphabetical relation, this implies $\mathrm{alph}(y_i)\times\mathrm{alph}(s_ju_j) \subseteq I$ for every $0 \leq i < j$. Moreover, we know $y_i \leq w_{i+1} = u_{i+1}y_{i+1}$, thus $y_i \leq y_{i+1}$, for every $i \geq 0$, since $\mathrm{alph}(y_i)\times\mathrm{alph}(u_{i+1}) \subseteq I$. Let us define finally v_i by $y_{i+1} = y_iv_{i+1}$ (and $v_0 = y_0$).

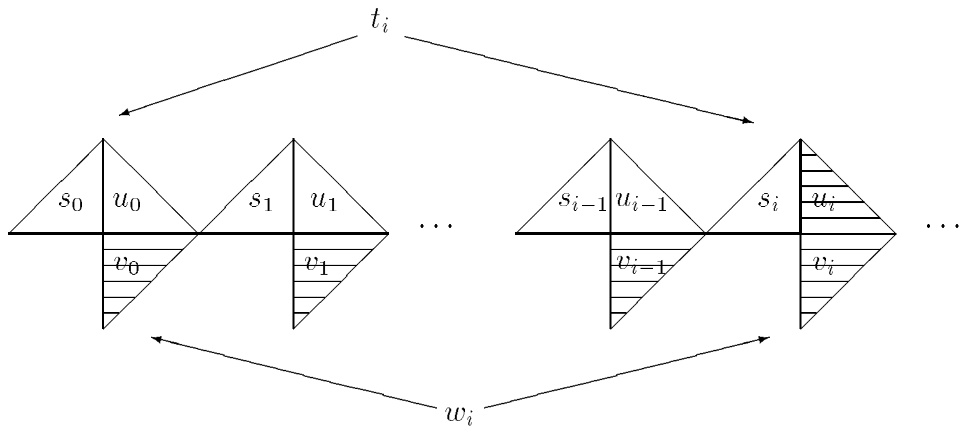

Figure 1: The factorization of $t_i w_i$, where w_i consists of the marked regions.

Summarizing our results we obtain $\mathrm{alph}(v_i) \times \mathrm{alph}(s_j u_j) \subseteq I$ for $j > i \geq 0$, and (see Fig. 1)

$$\begin{aligned} t_i &= t_0 x_0 \cdots x_{i-1} = s_0 u_0 s_1 \cdots u_{i-1} s_i \qquad (\text{with } s_0 := t_0), \\ w_i &= u_i y_i = u_i v_0 v_1 \cdots v_i. \end{aligned}$$

$\square$

Remark 3.9 By grouping factors together and appropriately renumbering we can even obtain the stronger independence relations $\mathrm{alph}(v_i) \times \mathrm{alph}(s_j u_j) \subseteq I$, for $i \geq 0$, $j > 0$. Yet we will not make use of this.

Example 3.10 Let $(\Sigma, D) = a \text{ --- } b \text{ --- } c \text{ --- } d$ and consider the sequences $(t_n)_{n \geq 0}, (w_n)_{n \geq 0} \subseteq \mathbb{M}$ defined by $t_n = c(ba)(b^2a) \cdots (b^n a)b$ and $w_n = d^{n+1} b^n a$. Then $x = \bigsqcup_{n \geq 0}(t_n w_n) = cd^\omega (ba)(b^2 a) \ldots$. With the notations of the preceeding proposition we have $s_0 = cb$, $s_i = b$ for $i \geq 1$, $u_i = b^i a$ and $v_i = d$ for $i \geq 0$.
For the following considerations it is worth noting that the sequence $(w_n)_{n \geq 0}$ is prefix-free. However, there is no recognizable, prefix free set K containing $(w_n)_{n \geq 0}$. Replacing the sequence $(w_n)_{n \geq 0}$ by the sequence $(w'_n)_{n \geq 0}$ with $w'_n = db^n a$ changes the v_i to $v'_0 = d$ and $v'_i = 1$ for $i \geq 1$. The following corollary states that $v'_0 \neq 1$ is possible due to $d \notin D(\mathrm{alphinf}(x))$.

Corollary 3.11 *Given $L, K \subseteq \mathbb{M}$ with $K \in \mathrm{Rec}(\mathbb{M})$ such that K is prefix free, i.e. $K\mathbb{M}_+ \cap K = \emptyset$. Let $(t_n)_{n \geq 0} \subseteq L$, $(w_n)_{n \geq 0} \subseteq K$ such that $\{\, t_n w_n \mid n \geq 0 \,\}$ is an infinite, directed set with $x = \bigsqcup\{\, t_n w_n \mid n \geq 0 \,\}$. Assume additionally $D(\mathrm{alph}(y)) \subseteq D(\mathrm{alphinf}(x))$, for every $y \in K$.*
Then there exist a subsequence of indices $(n_i)_{i \geq 0}$ 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

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

Proof. According to Prop. 3.8, there exist a subsequence of indices $(n_i)_{i\geq 0}$ 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 $x = \bigsqcup\{\, t_{n_i} w_{n_i} \mid i \geq 0\,\}$, $t_{n_i} = s_0 u_0 \cdots s_{i-1} u_{i-1} s_i$, $w_{n_i} = u_i v_0 \cdots v_{i-1} v_i$, and $\mathrm{alph}(v_i) \times \mathrm{alph}(s_j u_j) \subseteq I$, for $0 \leq i < j$. We show that the property of K of being prefix-free and the condition about $\mathrm{alphinf}(x)$ allow us to take $v_i = 1$ for every i.
Consider a morphism $\eta : \mathbb{M} \to S$ to a finite monoid S recognizing K and observe that we may assume without loss of generality that $\eta(u_i) = \eta(u_j)$ for all i, j (in order to achieve this, we may have to take a subsequence of indices $(m_i)_{i\geq 0}$ such that $\eta(u_{m_i}) = \eta(u_{m_j})$ for all i, j and then redefine s_{m_i} as $s_{m_{i-1}+1} u_{m_{i-1}+1} \cdots s_{m_i}$ and v_{m_i} as $v_{m_{i-1}+1} v_{m_{i-1}+2} \cdots v_{m_i}$, respectively).
Now, we have $u_i v_0 \cdots v_{i-2} v_{i-1} \leq w_{n_i} \in K$ for $i \geq 1$. Moreover, due to $\eta(u_i) = \eta(u_{i-1})$, it follows that $\eta(u_i v_0 \cdots v_{i-2} v_{i-1}) = \eta(u_{i-1} v_0 \cdots v_{i-2} v_{i-1}) = \eta(w_{i-1})$, hence $u_i v_0 \cdots v_{i-2} v_{i-1} \in K$ for $i \geq 1$. Therefore $v_i = 1$, since traces in K have no proper prefix in K. Additionally, since $\mathrm{alph}(v_0) \times \mathrm{alph}(s_i u_i) \subseteq I$ for every $i > 0$, we have $\mathrm{alph}(v_0) \cap D(\mathrm{alphinf}(x)) = \emptyset$. Hence, since $\mathrm{alph}(v_0) \subseteq \mathrm{alph}(w_i)$ with $w_i \in K$, $v_0 \neq 1$ would contradict the hypothesis $D(\mathrm{alphinf}(x)) \supseteq D(\mathrm{alph}(y))$ for every $y \in K$. Hence,

$$\begin{aligned} t_{n_i} &= s_0 u_0 \cdots s_{i-1} u_{i-1} s_i \quad \text{and} \\ w_{n_i} &= u_i. \end{aligned}$$

$\square$

The next proposition generalizes the corresponding result for infinitary word languages (see also [PP92]) 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 Σ^ω as a special case of the next proposition (recall Rem. 3.1).

Proposition 3.12 *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. The first inclusion is clear, since for every infinite trace $x = x_0 x_1 \ldots$ with $x_0 \in \mathbb{M}_s$ and $x_n \in \mathbb{M}_e$ for $n \geq 1$ we can choose some $x'_n \in \mathbb{P}_e$ such that $x'_n \leq x_n$, $n \geq 1$. Hence, $x = \bigsqcup\{\, x_0 x_1 \cdots x_{n-1} x'_n \mid n \geq 1\,\} \in \overrightarrow{\mathbb{M}_s \mathbb{P}_e} \cap \mathbb{R}_e$.
Consider now $x \in \overrightarrow{\mathbb{M}_s \mathbb{P}_e} \cap \mathbb{R}_e$, i.e., $D(\mathrm{alphinf}(x)) = D(\mathrm{alph}(e))$ and $x = \bigsqcup\{\, t_n w_n \mid n \geq 0, t_n \in \mathbb{M}_s, w_n \in \mathbb{P}_e\,\}$, with the right side being a directed set. By Corollary 3.11 we may assume

$$t_{n_i} = s_0 u_0 \cdots s_{i-1} u_{i-1} s_i \quad \text{and} \quad w_{n_i} = u_i,$$

for some index sequence $(n_i)_{i\geq 0}$ and trace sequences $(s_i)_{i\geq 0}, (u_i)_{i\geq 0} \subseteq \mathbb{M}$. In particular, we have $x = s_0 u_0 s_1 u_1 \cdots s_{i-1} u_{i-1} s_i \ldots$. We can apply Ramsey's Theorem in the usual way (e.g. as in [Gas91]) in order to obtain a factorization $(m_i)_{i\geq 0}$ of x,

$$x = (s_0 u_0 s_1 \cdots s_{m_0})\,(u_{m_0} s_{m_0+1} \cdots s_{m_1})\,(u_{m_1} s_{m_1+1} \cdots s_{m_2}) \ldots$$

with $\eta(u_{m_i} s_{m_i+1} \cdots s_{m_j}) =: f$ for every $0 \leq i < j$, and $\eta(s_0 u_0 \cdots s_{m_1}) =: s'$, hence $f \in E(S)$ and $s'f = s'$, since $s' \in Sf$. Obviously we have $s' = s$. Moreover, $f \in \eta(u_n)S = eS$ implies $f \leq e$, so $x \in \mathbb{M}_s \mathbb{M}_f^\omega$ with $f \leq e$. □

Corollary 3.13 *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. $\bigcup_{f\leq e} \mathbb{M}_s \mathbb{M}_f^\omega \quad = \quad \bigcup_{f \leq e} (\overrightarrow{\mathbb{M}_s \mathbb{P}_f} \cap \mathbb{R}_f)$.

Proof. (1) Let $s = e$ in Proposition 3.12, hence $sf = s$ means $ef = e$. Moreover, $f \leq e$ means $ef = f$, thus $e = f$ and

$$\mathbb{M}_e^\omega \quad \subseteq \quad \overrightarrow{\mathbb{M}_e \mathbb{P}_e} \cap \mathbb{R}_e \quad \subseteq \quad \mathbb{M}_e^\omega.$$

(2) We know for $sf = s$ and $f \in E(S)$:

$$\mathbb{M}_s \mathbb{M}_f^\omega \quad \subseteq \quad \overrightarrow{\mathbb{M}_s \mathbb{P}_f} \cap \mathbb{R}_f \quad \subseteq \quad \bigcup_{h \leq f} \mathbb{M}_s \mathbb{M}_h^\omega,$$

hence,

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

with the last inclusion due to the transitivity of $\leq$. □

The following lemma expresses a property of the set $P_L = \{\,(s,e) \in S^2 \mid e \in E(S), se = s \text{ and } \mathbb{M}_s \mathbb{M}_e^\omega \cap L \neq \emptyset\,\}$ associated to $L \in \mathrm{Rec}(\mathbb{R})$. Two pairs $(s,e), (s',e') \in S^2$ are called *conjugated* ([PP92]) if for some $x, y \in S$ the equalities $s' = sx$, $e = xy$ and $e' = yx$ hold.

Lemma 3.14 *Let $L = \bigcup_{(s,e)\in P_L} \mathbb{M}_s \mathbb{M}_e^\omega$ be recognized by a morphism $\eta : \mathbb{M} \to S$ onto a finite monoid S, with the set P_L defined as above. Then for all pairs $(s,e), (s',e') \in P_L$ the following holds:*

$$\mathbb{M}_s \mathbb{M}_e^\omega \;\cap\; \mathbb{M}_{s'} \mathbb{M}_{e'}^\omega \neq \emptyset \quad \textit{if and only if} \quad (s,e) \textit{ and } (s',e') \textit{ are conjugated.}$$

Proof. The direction from right to left is clear due to the definition of conjugated pairs and to η being surjective. Let us consider $t \in \mathbb{M}_s\mathbb{M}_e^\omega \cap \mathbb{M}_{s'}\mathbb{M}_{e'}^\omega$, i.e., there exist $t_0 \in \mathbb{M}_s$, $(t_n)_{n\geq 1} \subseteq \mathbb{M}_e$, $t'_0 \in \mathbb{M}_{s'}$, $(t'_n)_{n\geq 1} \subseteq \mathbb{M}_{e'}$ with $t = t_0 t_1 \ldots = t'_0 t'_1 \ldots$. Since $e, e' \in E(S)$ we may group together consecutive t_n's (t'_n's) and assume for every $n \geq 0$:

$$t'_0 t'_1 \cdots t'_n \leq t_0 t_1 \cdots t_n \leq t'_0 t'_1 \cdots t'_{n+1}$$

Let us define 2 sequences of finite traces $(s_n)_{n\geq 0}, (u_n)_{n\geq 0}$ by setting:

$$\begin{aligned} t_0 t_1 \cdots t_n &= t'_0 t'_1 \cdots t'_n s_n, \\ t'_0 t'_1 \cdots t'_{n+1} &= t_0 t_1 \cdots t_n u_n \end{aligned}$$

This yields the following equations, due to $\mathbb{M}$ being left-cancellative:

$$t_{n+1} = u_n s_{n+1}, \qquad t'_{n+1} = s_n u_n \quad \text{and} \quad t = t'_0 s_0 u_0 s_1 u_1 \ldots,$$

for every $n \geq 0$ ($t_0 = t'_0 s_0$).
Let now $0 \leq i < j$ be such that $\eta(s_i) = \eta(s_j) =: x$ (S is finite), and let additionally $y := \eta(u_i s_{i+1} \cdots u_{j-1})$. The following relations show that (s, e) and (s', e') are conjugated pairs:

$$\begin{aligned} s &= \eta(t_0 t_1 \cdots t_i) = \eta(t'_0 t'_1 \cdots t'_i)\,\eta(s_i) = s'x, \\ e &= \eta(u_i s_{i+1} \cdots u_{j-1} s_j) = yx, \\ e' &= \eta(s_i u_i \cdots s_{j-1} u_{j-1}) = xy. \end{aligned}$$

□

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 languages $\{\overrightarrow{L} \cap \mathbb{R}_A \mid L \in \mathrm{Rec}(\mathbb{M}), A \subseteq \Sigma\}$. We obtain the well-known result for words as a special case of this theorem.

Theorem 3.15 *Let $L \in \mathrm{Rec}(\mathbb{R})$ be recognized by $\eta : \mathbb{M} \to S$ onto a finite monoid S, 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 $f \in E(S)$. Since (s,e) and (s,f) are conjugated with $x = e$ and $y = f$ ($f = xy, e = yx, s = sx$), by Lemma 3.14 and Rem. 2.2 we obtain $(s,f) \in P_L$. 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

$$\mathbb{M}_s\mathbb{M}_e^\omega \;\cap\; \mathbb{M}_s\mathbb{M}_f^\omega = \emptyset. \tag{1}$$

Otherwise, (s,e) and (s,f) would be conjugated (by Lemma 3.14) so there would exist some $x, y \in S$ such that $sx = s, e = xy, f = yx$. This would imply for $n \geq 0$:

$$e = e^2 = xyxy = xfy \overset{f \leq e}{=} xefy = x^n e(fy)^n$$

and, since S is finite, there exist $n, q \in \mathbb{N}$, $q \neq 0$, with $z^{n+q} = z^n$ for every $z \in S$. In this case, we would obtain $e \in fS$, since $e = x^n e(fy)^{n+q} = e(fy)^q = ef(yf)^{q-1}y \overset{ef=f}{\in} fS$. This means $f \not< e$, thus leading to a contradiction.
We may now conclude the proof:

$$\begin{aligned}
\bigcup_{f\mathcal{R}e} \mathbb{M}_s\mathbb{M}_f^\omega &= \bigcup_{f \leq e, f \not< e} \mathbb{M}_s\mathbb{M}_f^\omega \\
&\overset{(1)}{=} \bigcup_{f \leq e} \mathbb{M}_s\mathbb{M}_f^\omega \setminus \bigcup_{f < e} \mathbb{M}_s\mathbb{M}_f^\omega \\
&= \bigcup_{f \leq e} \mathbb{M}_s\mathbb{M}_f^\omega \setminus \bigcup_{f < e} \bigcup_{g \leq f} \mathbb{M}_s\mathbb{M}_g^\omega \\
&\overset{\text{Cor.3.13}}{=} \bigcup_{f \leq e} (\overrightarrow{\mathbb{M}_s\mathbb{P}_f} \cap \mathbb{R}_f) \setminus \bigcup_{f < e} \bigcup_{g \leq f} (\overrightarrow{\mathbb{M}_s\mathbb{P}_g} \cap \mathbb{R}_g) \\
&= \bigcup_{f \leq e} (\overrightarrow{\mathbb{M}_s\mathbb{P}_f} \cap \mathbb{R}_f) \setminus \bigcup_{f < e} (\overrightarrow{\mathbb{M}_s\mathbb{P}_f} \cap \mathbb{R}_f)
\end{aligned}$$

□

Corollary 3.16 Rec($\mathbb{R}$) *is equivalent to the Boolean closure of the family of real trace languages of type* $\overrightarrow{L} \cap \mathbb{R}_A$*, where* $L \in \text{Rec}(\mathbb{M})$ *and* $A \subseteq \Sigma$*.*

Proof. Every language $\overrightarrow{L} \cap \mathbb{R}_A$, $L \in \text{Rec}(\mathbb{M})$ and $A \subseteq \Sigma$, is recognizable and Rec($\mathbb{R}$) is closed under Boolean operations. The other direction is Theorem 3.15.
□

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 $L \in \text{Rec}(\mathbb{M})$. Since for every $\mathbb{R}_A$ ($A \subseteq \Sigma$) one can clearly exhibit a deterministic asynchronous cellular Muller

automaton (see Example 2.3), we will obtain at the end the equivalence between $\mathrm{Rec}(\mathbb{R})$ and the family of languages accepted by deterministic asynchronous cellular Muller automata. Recall also that the family of languages accepted by asynchronous cellular Muller automata is closed under 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 4.1 *Let $A = \bigcup_{i=1}^{k} A_i$ be a partition in connected components for $A \subseteq \Sigma$, i.e.,*

$$\begin{array}{ll} A_i \times A_j \subseteq I & \text{for } i \neq j \text{ and} \\ A_i \text{ is connected for every } i = 1, \ldots, k. & \end{array}$$

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 $\mathbb{M}$:

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

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

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

Proof. The proof is similar to the proof of Proposition 3.12. Let us consider $u \in \overrightarrow{L}$, $\mathrm{alphinf}(u) = A = \bigcup_{i=1}^{k} A_i$. Then there exists a factorization of u, $u = u_0 u_1 u_2 \ldots$ with the property that $\mathrm{alph}(u_n) = A$ for every $n \geq 1$ and $u_0 u_1 \cdots u_n \in L$ for all $n \geq 0$. Furthermore we write each u_n as a product $u_n = u_{n,1} \cdots u_{n,k}$, where $\mathrm{alph}(u_{n,i}) = A_i$ for every $1 \leq i \leq k$. For each $n \geq 1$ and $1 \leq i \leq k$ consider the sequence $u_{n,i} u_{n+1,i} \ldots$. Due to A_i being connected, this sequence has a least prefix $u'_{n,i}$ such that $\max(u'_{n,i}) = \{a_i\}$ and $\mathrm{alph}(u'_{n,i}) = A_i$. We define thus k sequences $(u'_{n,i})_{n\geq 1} \subseteq \mathbb{P}_{A,i}$ such that for every $n \geq 1$, $u_0 \cdots u_{n-1} u'_{n,1} \cdots u'_{n,k} \in L\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}$ and $u = \bigsqcup \{\, u_0 \cdots u_{n-1} u'_{n,1} \cdots u'_{n,k} \mid n \geq 1\,\}$. Hence, $u \in \overrightarrow{L\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}} \cap \mathrm{Inf}(A)$.
For the converse direction, consider $x = \bigsqcup \{\, t_n w_n \mid t_n \in L,\ w_n \in \mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k},\ n \geq 0\,\}$ with $\mathrm{alphinf}(x) = A$. By Corollary 3.11 (note that $\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k} \in \mathrm{Rec}(\mathbb{M})$ and $\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}\mathbb{M}_{+} \cap \mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k} = \emptyset$) we obtain for a subsequence of indices $(n_i)_{i\geq 0}$ with $x = \bigsqcup_{i\geq 0} t_{n_i} w_{n_i}$:

$$\begin{aligned} t_{n_i} &= s_0 u_0 s_1 \cdots s_{i-1} u_{i-1} s_i \ \in\ L \\ w_{n_i} &= u_i \ \in\ \mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}, \end{aligned}$$

for some sequences of finite traces $(s_i)_{i\geq 0}, (u_i)_{i\geq 0}$.
Hence, $x = s_0u_0s_1u_1\ldots$, with $s_0u_0\cdots s_i \in L$ for every $i \geq 0$ and thus $x \in \overrightarrow{L}$. □

Before stating the main theorem of this section, let us recall some important concepts of the construction of Zielonka for asynchronous cellular automata 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 $t \in \mathbb{M}$,

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

Zielonka's construction is based on the concept of *asynchronous mapping* (see e.g. [Die90]). It is a mapping $\mu : \mathbb{M} \to Q$ 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 $\mu(\partial_{A\cup B}(t))$ is uniquely determined by $\mu(\partial_A(t))$ and $\mu(\partial_B(t))$.
- The value of $\mu(\partial_a(ta))\,(= \mu(\partial_{D(a)}(ta)))$ is uniquely determined by a and $\mu(\partial_{D(a)}(t))$.

Given an asynchronous mapping $\mu : \mathbb{M} \to Q$ 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 $\mu^{-1}(R)$:

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

(It is easy to check that δ is well-defined and satisfies the requirement for the partially defined transition function of an asynchronous cellular automaton, see e.g. [Die90]). The initial state of $\mathcal{A}$ is $q_0 = (\mu(1))_{a\in\Sigma}$ and F is given by $F = \{\, (\mu(\partial_a(t)))_{a\in\Sigma} \mid \mu(t) \in R \,\}$. Moreover, we have $\delta(q_0, t) = (\mu(\partial_a(t)))_{a\in\Sigma}$.
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 the basic step in the construction of Zielonka of an asynchronous cellular automaton accepting K is to provide an asynchronous mapping $\mu : \mathbb{M} \to Q$ to a finite set Q, such that the morphism η factorizes through μ, i.e. $\eta = \pi \circ \mu$ for a mapping $\pi : Q \to S$. The asynchronous cellular automaton is then defined as above, with $R = \pi^{-1}(\eta(K))$.
A crucial feature of the construction is the fact that for any finite trace t, the local states reached by the resulting asynchronous cellular automaton after having read t, which correspond to the maximal elements of t, already determine whether t is being accepted or not. This follows with $t = \partial_{\max(t)}(t)$, hence $\mu(t) = \mu(\partial_{\max(t)}(t))$, which means that $\mu(t)$ is exactly determined by $\{\, \mu(\partial_a(t)) \mid a \in \max(t) \,\}$, since μ is an asynchronous mapping.

Theorem 4.2 *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

$$\overrightarrow{L} = \bigcup_{A \subseteq \Sigma} (\overrightarrow{L} \cap \mathrm{Inf}(A)) \stackrel{4.1}{=} \bigcup_{A \subseteq \Sigma} (\overrightarrow{L\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}} \cap \mathrm{Inf}(A)),$$

with $\mathbb{P}_{A,1}, \dots \mathbb{P}_{A,k}$ depending on A and defined as in Lemma 4.1, it will suffice to construct a deterministic asynchronous Muller automaton accepting the language $\overrightarrow{L\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}} \cap \mathrm{Inf}(A)$, for $A \neq \emptyset$.
Let $\eta : \mathbb{M} \to S$ be a morphism to a finite monoid recognizing $L\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}$, and let $\mu : \mathbb{M} \to Q$ be the asynchronous mapping to the finite set Q such that there exists a mapping π with $\eta = \pi \circ \mu$. Finally, consider the deterministic asynchronous cellular automaton $\mathcal{A}' = ((Q'_a)_{a\in\Sigma}, (\delta'_a)_{a\in\Sigma}, q'_0, F)$ accepting $L\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}$, which is obtained by Zielonka's construction. In particular, we have $\delta'(q'_0, t) = (\mu(\partial_a(t)))_{a\in\Sigma}$, for every $t \in \mathbb{M}$, $a \in \Sigma$.
Furthermore, consider for every $f \in F$ the language

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

Clearly $\overrightarrow{L\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}} \cap \mathrm{Inf}(A) = \bigcup_{f\in F} L_{A,f}$ 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 $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, \emptyset, \mathcal{T})$ such that $L_{A,f} \subseteq L(\mathcal{A}) \subseteq \overrightarrow{L\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}} \cap \mathrm{Inf}(A)$. Let us define for $a \in \Sigma$:

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

The $\mathbb{Z}/2\mathbb{Z}$ component of the local states of $\mathcal{A}$ is used in order to distinguish whether a letter belongs or not to the alphabet at infinity, since we have $\delta_a(s_{D(a)}) \neq s_a$, for every $s_a \in Q_a$, $a \in \Sigma$. Recall that $\mathrm{inf}_a(t)$ denotes the set of local a-states which occur infinitely often on the computation path labelled by t (if existent). 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 $\mathcal{T}$:

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

The inclusion $L_{A,f} \subseteq L(\mathcal{A})$ is not hard to be seen.

Now 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 $(i_a)_{a\in\Sigma} \in (\mathbb{Z}/2\mathbb{Z})^{\Sigma}$:

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

(such a factorization exists because of the definition of the table $\mathcal{T}$ together with $a_i \in A_i$, with $A_i \times A_j \subseteq I$, for $i \neq j$).
We show that $t_0 t_1 \cdots t_n \in L\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}$, $n \geq 0$. Evidently, $\delta'(q_0', t_0 \cdots t_n)_a = f_a$, for $a \in (\Sigma \setminus A) \cup \{a_1, \ldots, a_k\}$, $n \geq 0$. From the definition of $\mathcal{A}'$, we know $f = (\mu(\partial_a(u)))_{a\in\Sigma}$ for some $u \in L\mathbb{P}_{A,1} \cdots \mathbb{P}_{A,k}$. 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:

$$\delta'(q_0', t_0 \cdots t_n)_a \quad = \quad \mu(\partial_a(t_0 \cdots t_n)) \quad = \quad \mu(\partial_a(u)) \quad = \quad f_a$$

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

Remark 4.3 Note that our result above differs from the situation encountered in the case of words, where the languages $\overrightarrow{L}$, with $L \subseteq \Sigma^*$ recognizable, are precisely those recognized by deterministic Büchi automata (therefore called deterministic languages). Here we need the more powerful Muller automata, due to the crucial knowledge of the alphabet occurring infinitely often.

We can now state the generalization of McNaughton's theorem for infinitary trace languages, which is the main result of the paper.

Corollary 4.4 *The family of recognizable real trace languages is equivalent to the family of languages, which are accepted by deterministic asynchronous cellular Muller automata.*

Proof. The inclusion from left to right is due to Theorems 3.15, 4.2. The other way round, if $L \subseteq \mathbb{R}$ is accepted by a (deterministic) asynchronous cellular Muller automaton then L can be expressed as a Boolean combination of languages accepted by Büchi asynchronous cellular automata. $\square$

Due to asynchronous automata being automatically I-diamond, we immediately obtain:

Corollary 4.5 *Every recognizable, closed infinitary word language can be accepted by a deterministic I-diamond Muller automaton.*

5 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 Thm. 3.15 can be improved, i.e. by showing that every recognizable real trace language is already a Boolean combination of languages $\overrightarrow{L}$ with $L \in \mathrm{Rec}(\mathbb{M})$. The question, which class of deterministic I-diamond Muller automata accepts closed languages will constitute the subject of a forthcoming paper.

Acknowledgements: We thank Paul Gastin and Jean-Eric Pin for fruitful discussions and Wiesław Zielonka who told us that independently he has developed similar ideas. We also thank the anonymous referees for many valuable comments which helped improving the readability of this paper, in particular concerning the proof of Proposition 3.8.

References

[Arn85] A. Arnold. A syntactic congruence for rational ω-languages. *Theoret. Comput. Sci.*, 39:333–335, 1985.

[CF69] P. Cartier and D. Foata. *Problèmes combinatoires de commutation et réarrangements*. Number 85 in Lecture Notes in Mathematics. Springer, Berlin-Heidelberg-New York, 1969.

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

[CP85] R. Cori and D. Perrin. Automates et commutations partielles. *R.A.I.R.O. — Informatique Théorique et Applications*, 19:21–32, 1985.

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

[Die91] V. Diekert. On the concatenation of 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 105–117, Berlin-Heidelberg-New York, 1991. Springer. To appear 1993 in Theoret. Comput. Sci.

[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, Berlin-Heidelberg-New York, 1991. Springer.

[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.

[GPZ91] P. Gastin, A. Petit, and W. Zielonka. A Kleene theorem for infinite trace languages. In J. Leach Albert et al., editors, *Proceedings of the 18th International Colloquium on Automata Languages and Programming (ICALP'91), Madrid (Spain) 1991*, number 510 in Lecture Notes in Computer Science, pages 254–266, Berlin-Heidelberg-New York, 1991. Springer.

[GR91] P. Gastin and B. Rozoy. The poset of infinitary traces. Tech. Rep. LITP 91.07, Université Paris 6 (France), 1991. To appear in *Theoret. Comp. Sci.*

[Kwi90] M. Kwiatkowska. A metric for traces. *Information Processing Letters*, 35:129–135, 1990.

[Maz77] A. Mazurkiewicz. Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University, Aarhus, 1977.

[Maz87] A. Mazurkiewicz. Trace theory. In W. Brauer et al., editors, *Petri Nets, Applications and Relationship to other Models of Concurrency*, number 255 in Lecture Notes in Computer Science, pages 279–324, Berlin-Heidelberg-New York, 1987. Springer.

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

[PP92] D. Perrin and J.E. Pin. Mots Infinis. Tech. Rep. LITP 92.17, Université Paris 6 (France), 1992. 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, Berlin-Heidelberg-New York, 1989. Springer.