Classifying Recognizable Infinitary Trace Languages Using Word Automata

Namit Chaturvedi* and Marcus Gelderie**

RWTH Aachen Univeristy, Lehrstuhl für Informatik 7, Aachen, Germany
{chaturvedi,gelderie}@automata.rwth-aachen.de

Abstract. We address the problem of providing a Borel-like classification of languages of infinite Mazurkiewicz traces, and provide a solution in the framework of ω-automata over infinite words – which is invoked via the sets of linearizations of infinitary trace languages. We identify trace languages whose linearizations are recognized by deterministic weak or deterministic Büchi (word) automata. We present a characterization of the class of linearizations of all recognizable ω-trace languages in terms of Muller (word) automata. Finally, we show that the linearization of any recognizable ω-trace language can be expressed as a Boolean combination of languages recognized by our class of deterministic Büchi automata.

1 Introduction

Traces were introduced as models of concurrent behaviors of distributed systems by Mazurkiewicz, who later also provided an explicit definition of infinite traces [7]. Zielonka demonstrated the close relation between traces and trace-closed sets of words, which can be viewed as "linearizations" of traces, and established automata-theoretic results regarding recognizability of languages of finite traces [11] (cf. also [3,6]). Later, [4,2,8] enriched the theory of recognizable languages of infinite traces (*recognizable ω-trace languages*), by introducing models of computations viz. asynchronous Büchi automata and deterministic asynchronous Muller automata. Being closely related to word languages, a set of infinite traces is recognizable iff the corresponding trace-closed set of infinite words is.

In the case of ω-regular word languages, there exists a straightforward characterization of languages recognized by deterministic Büchi automata, and a result due to Landweber states that it is decidable whether a given ω-regular language is deterministically Büchi recognizable [9, Chapter 1]. However, analogous results over recognizable ω-trace languages have only recently been established in terms of "synchronization-aware" asynchronous automata [1].

While asynchronous automata are useful in implementing distributed monitors and distributed controllers, their constructions are prohibitively expensive even by automata-theoretic standards. On the other hand, for applications like

* Supported by the DFG Research Training Group-1298 AlgoSyn, and the CASSTING Project funded by the European Commission's 7th Framework Programme.

** Supported by the DFG Research Training Group-1298 AlgoSyn.

E. Csuhaj-Varjú et al. (Eds.): MFCS 2014, Part I, LNCS 8634, pp. 171–182, 2014.

model-checking and formal verification, word automata recognizing trace-closed languages would already allow for analysis of most of the interesting properties pertaining to distributed computations.

Therefore, in this paper, we study classes of ω-regular word languages that allow us to "transfer" interesting results to the corresponding classes of recognizable ω-trace languages. In particular, motivated by the Borel hierarchy for regular languages of infinite words, our main contribution is a new setup for a classification theory for recognizable ω-trace languages in terms of trace-closed, ω-regular word languages.

Recall that in the sequential setting, reachability languages and deterministically Büchi recognizable languages – constituting the lowest levels of the Borel hierarchy – can be obtained via natural operations over regular languages $K \subseteq \Sigma^*$ in the following ways:

- $\mathsf{ext}(K) = K \cdot \Sigma^\omega = \{\alpha \in \Sigma^\omega \mid \alpha \text{ has a prefix in } K\}$
- $\mathsf{lim}(K) = \{\alpha \in \Sigma^\omega \mid \alpha \text{ has infinitely many prefixes in } K\}$

These operations, which we call the *infinitary extension* and the *infinitary limit* of K, can be generalized to obtain infinitary extensions $\mathsf{ext}(T)$ and infinitary limits $\mathsf{lim}(T)$ of regular trace languages T.

In this paper, given the trace-closed word language K corresponding to a regular trace language T, we firstly show that K can be modified to K_I such that $\mathsf{ext}(K_I)$ is also trace-closed and corresponds to the linearization of $\mathsf{ext}(T)$ (here I denotes the independence relation over the alphabet Σ). Building on this, we are able to characterize the class of Boolean combinations of languages $\mathsf{ext}(T)$ as precisely those whose linearizations are recognized by the class of "I-diamond" deterministic weak automata (DWAs).

Next, we consider infinitary limits. Here the situation is different, in that there exist regular trace languages T such that although the trace-closed word language L corresponding to $\mathsf{lim}(T)$ is ω-regular, it is not recognized by any I-diamond deterministic Büchi automaton (DBA). We therefore introduce the class of *limit-stable* word languages K – and by extension limit-stable trace languages T – such that the correspondence of Fig. 1b holds, and $\mathsf{lim}(K)$ can be characterized in terms of I-diamond DBA.

It is well known that every trace-closed ω-regular language is recognized by an I-diamond Muller automata [2]. We characterize these languages in terms of a well defined class of I-diamond Muller automata. And lastly, justifying our

(a) The correspondence of infinitary extensions for all regular trace languages T.

(b) The correspondence of infinitary limits for all *limit-stable* trace languages T.

Fig. 1. From trace-closed regular languages to trace-closed ω-regular languages

definitions, we show that every trace-closed ω-regular word language (that is, the linearization of any recognizable ω-trace language) can be expressed as a finite Boolean combination of languages $\mathsf{lim}(K)$, with K limit-stable.

In related work, Diekert & Muscholl [2] consider a form of "deterministic" trace languages. It is shown that every recognizable language of infinite traces is a Boolean combination of these deterministic languages. However, in the attempt to characterize the corresponding "deterministic" trace-closed word languages in terms of I-diamond automata, it is necessary to extend the Büchi acceptance condition beyond what we know from standard definitions [8]. It had been left open in [8] whether there exists a class of deterministic asynchronous Büchi automata for deterministic trace languages.

We begin with presenting the basic definitions and notations. In Sec. 3, we present the operations ext and lim that allow for construction of recognizable ω-trace languages from regular trace languages. In particular, we exhibit recognizable ω-trace languages whose linearizations are recognized by I-diamond DWA, and those whose linearizations are I-diamond DBA recognizable. Finally, we establish our main result demonstrating the expressiveness of I-diamond DBA recognizable trace-closed languages.

2 Preliminaries

We denote a recognizable language of finite words, or simply a *regular language*, with the upper case letter K and a class of such languages with $\mathcal{K}$. Finite words are denoted with lower case letters u, v, w etc. Infinite words are denoted by lower case Greek letters α and β, and a recognizable language of infinite words, or simply an *ω-regular language*, by upper case L. For a word u or α, we denote its infix starting at position i and ending at position j by $u[i,j]$ or $\alpha[i,j]$, and the i^{th} letter with $u[i]$ or $\alpha[i]$. For a language K, we define $\overline{K} := \Sigma^* \setminus K$.

We assume the reader is familiar with the notions of Deterministic Finite Automata (DFAs) and Deterministic Büchi Automata (DBAs). We say that a language is *DBA recognizable* if it is recognized by a DBA. For the class REG of regular languages, the class lim(REG) coincides with the DBA recognizable languages. Further, the class BC(lim(REG)) of finite Boolean combinations of languages from lim(REG) is also the class of ω-regular languages, and it coincides with the class of languages recognized by nondeterministic Büchi or deterministic Muller automata.

Recall that a Deterministic Weak Automaton (DWA) is a DBA where every strongly connected component of the transition graph has only accepting states or only rejecting states. For a regular language K, the minimal DFA recognizing K also recognizes $\mathsf{lim}(K)$ as a DBA. Given the minimal DFA $\mathfrak{A} = (Q, \Sigma, q_0, \delta, F)$ recognizing K, a DWA $\mathfrak{A}' := (Q', \Sigma, q_0, \delta', F')$ recognizing $\mathsf{ext}(K)$, respectively $\overline{\mathsf{ext}(K)}$, can be constructed as follows:

1. For a symbol $\bot \notin Q$ and define $Q' := (Q \setminus F) \cup \{\bot\}$.
2. For each $q \in Q', a \in \Sigma$, define $\delta'(q,a) := \begin{cases} \delta(q,a) & \text{if } q \neq \bot \text{ and } \delta(q,a) \notin F, \\ \bot & \text{otherwise.} \end{cases}$

3. Define $F' := \{\bot\}$, respectively $F' := Q' \setminus \{\bot\}$.

The family of DWAs is closed under Boolean operations. For an ω-language L, define a congruence $\sim_L \subseteq \Sigma^* \times \Sigma^*$ where $u \sim_L v \Leftrightarrow \forall \alpha \in \Sigma^\omega, u\alpha \in L$ iff $v\alpha \in L$. If L is recognized by a DWA then this congruence has a finite index. We say that an ω-language is *weakly recognizable* if it is recognized by a DWA. The class BC(ext(REG)) of finite Boolean combinations of languages in ext(REG) is exactly the set of weakly recognizable languages [10].

Remark 1 (The minimal DWA [5]). For a weakly recognizable language L, if M is the index of the congruence defined above, then L is recognized by a DWA $\mathfrak{A} = (Q, \Sigma, q_0, \delta, F)$ with $|Q| = M$. Also, for every $q \in Q$ there exists a word $u_q \in \Sigma^*$ such that for each $u \in \Sigma^*, \delta(q_0, u) = q$ iff $u \in [u_q]_{\sim_L}$. ⊠

Turning to traces, let $I \subseteq \Sigma \times \Sigma$ denote an irreflexive[1], symmetric *independence* relation over an alphabet Σ, then $D := \Sigma^2 \setminus I$ is the reflexive, symmetric *dependence* relation over Σ. We refer to the pair (Σ, I) as the *independence alphabet.* For any letter $a \in \Sigma$, we define $I_a := \{b \in \Sigma \mid aIb\}$ and $D_a := \{b \in \Sigma \mid aDb\}$. A *trace* can be identified with a labeled, acyclic, directed *dependence graph* $[V, E, \lambda]$ where V is a set of countably many vertices, $\lambda\colon V \to \Sigma$ is a labeling function, and E is a countable set of edges such that, firstly, for every $v_1, v_2 \in V\colon \lambda(v_1)D\lambda(v_2) \Leftrightarrow (v_1, v_2) \in E \vee (v_2, v_1) \in E$; secondly, every vertex has only finitely many predecessors. $\mathbb{M}(\Sigma, I)$ and $\mathbb{R}(\Sigma, I)$ represent the sets of all finite and infinite traces whose dependence graphs satisfy the two conditions above. We denote finite traces with the letter t, and an infinite trace with θ; the corresponding languages with T and Θ respectively. For a trace $t = [V, E, \lambda]$, define $\mathrm{alph}(t) := \{a \in \Sigma \mid \emptyset \neq \lambda^{-1}(a) \subseteq V\}$, and similarly for a trace θ. For an infinite trace, define $\mathrm{alphinf}(\theta) := \{a \in \Sigma \mid |\lambda^{-1}(a)| = \infty\}$.

For two traces t_1, t_2, $t_1 \sqsubseteq t_2$ (or $t_1 \sqsubset t_2$) denotes that t_1 is a (proper) prefix of t_2. We denote the prefix relation between words similarly. The least upper bound of two finite traces, whenever it exists, denoted $t_1 \sqcup t_2$ is the smallest trace s such that $t_1 \sqsubseteq s$ and $t_2 \sqsubseteq s$. Whenever it exists, one can similarly refer to the least upper bound $\bigsqcup S$ of a finite or an infinite set S of traces. The concatenation of two traces is denoted as $t_1 \odot t_2$. Note that for any t, θ the concatenation $t \odot \theta \in \mathbb{R}(\Sigma, I)$. However, $\theta \odot t \in \mathbb{R}(\Sigma, I)$ iff $\mathrm{alphinf}(\theta) I \mathrm{alph}(t)$.

The canonical morphism $\Gamma\colon \Sigma^* \to \mathbb{M}(\Sigma, I)$ associates finite words with finite traces, and the inverse mapping $\Gamma^{-1}\colon \mathbb{M}(\Sigma, I) \to 2^{\Sigma^*}$ associates finite traces with equivalence classes of words. The morphism Γ can also be extended to a mapping $\Gamma\colon \Sigma^\omega \to \mathbb{R}(\Sigma, I)$. For a (finite or infinite) trace t, the set $\Gamma^{-1}(t)$ represents the *linearizations* of t. Two words u, v are equivalent, denoted $u \sim_I v$, iff $\Gamma(u) = \Gamma(v)$. We note that for finite traces the relation $\sim_I$ coincides with the reflexive, transitive closure of the relation $\{(uabv, ubav) \mid u, v \in \Sigma^* \wedge aIb\}$. For a word w, define the set $[w]_{\sim_I} := \Gamma^{-1}(\Gamma(w))$. Finally, we say that a word language K is *trace-closed* iff $K = [K]_{\sim_I}$, where $[K]_{\sim_I} := \bigcup_{u \in K} [u]_{\sim_I}$.

[1] A relation R is *irreflexive* if for no x we have xRx.

Definition 2. *A trace language $T \subseteq \mathbb{M}(\Sigma, I)$ (resp. $\Theta \subseteq \mathbb{R}(\Sigma, I)$) is called* recognizable *iff $\Gamma^{-1}(T)$ (resp. $\Gamma^{-1}(\Theta)$) is a recognizable word language.*

We denote the classes of recognizable languages of finite and infinite traces with $\mathsf{Rec}(\mathbb{M}(\Sigma, I))$ and $\mathsf{Rec}(\mathbb{R}(\Sigma, I))$ respectively.

Asynchronous cellular automata have been introduced [2,4] as acceptors of recognizable ω-trace languages. However, a global view of their (local) transition relations yields a notion of automata that recognize trace-closed word languages. Throughout this paper, we take this global view of asynchronous automata. Formally, a *deterministic asynchronous cellular automaton (DACA)* over (Σ, I) is a 4-tuple $\mathfrak{a} = (\prod_{a\in\Sigma} Q_a, (\delta_a)_{a\in\Sigma}, q_0, F)$, consisting of sets Q_a of *local states* for each letter $a \in \Sigma$, and where $q_0 \in \prod_{a\in\Sigma} Q_a$, $\delta_a \colon \prod_{b\in D_a} Q_b \to Q_a$ and $F \subseteq \prod_{a\in\Sigma} Q_a$. Given a state $q \in \prod_{a\in\Sigma} Q_a$ and a letter $b \in \Sigma$, the unique b-successor $\delta(q, b) = q' = (q'_a)_{a\in\Sigma} \in \prod_{a\in\Sigma} Q_a$ is given by $q'_b = \delta_b((q_a)_{a\in D_b})$ and $q'_a = q_a$ for all $a \neq b$. That is, the only component that changes its state is the component corresponding to b. Given a word $u \in \Sigma^*$ the *run* ρ_u of $\mathfrak{a}$ on u is given as usual by $\rho_u(0) = q_0$ and $\rho_u(i+1) = \delta(\rho_u(i), u[i])$. This definition extends naturally to infinite runs ρ_α on infinite $\alpha \in \Sigma^\omega$. Define $\mathsf{occ}_a(\rho)$ of (a finite or an infinite) run ρ to be the set $\{\rho(0)_a, \rho(1)_a, \ldots\} \subseteq Q_a$. Likewise, $\mathsf{inf}_a(\rho) = \{q \in Q_a \mid \exists^\infty n \colon \rho(n)_a = q\}$.

A *deterministic asynchronous cellular Muller automaton* [2] (a *DACMA*) is an asynchronous automaton $\mathfrak{a} = (\prod_{a\in\Sigma} Q_a, (\delta_a)_{a\in\Sigma}, q_0, \mathcal{F})$ with the acceptance table $\mathcal{F} \subseteq \prod_{a\in\Sigma} \mathscr{P}(Q_a)$, where $\mathscr{P}(Q_a)$ denotes the power set of Q_a. A DACMA *accepts* $\alpha \in \Sigma^\omega$ if for some $F = (F_a)_{a\in\Sigma} \in \mathcal{F}$ we have $\forall a \in \Sigma \colon \mathsf{inf}_a(\rho_\alpha) = F_a$. A *deterministic asynchronous cellular Büchi automaton* (a *DACBA*) is a tuple $\mathfrak{a} = (\prod_{a\in\Sigma} Q_a, (\delta_a)_{a\in\Sigma}, q_0, \mathcal{F})$, $F \subseteq \prod_{a\in\Sigma} \mathscr{P}(Q_a)$. A DACBA *accepts* $\alpha \in \Sigma^\omega$ if for some $F = (F_a)_{a\in\Sigma} \in \mathcal{F}$ we have $F_a \subseteq \inf_a(\rho_\alpha)$.

While it is known that the class of DACMAs characterize precisely the class of recognizable ω-trace languages [2], no such correspondence is known for the class of languages recognized by DACBAs [8].

A word automaton $\mathfrak{A} = (Q, \Sigma, q_0, \delta)$ is called *I-diamond* if for every $(a, b) \in I$ and every state $q \in Q$, $\delta(q, ab) = \delta(q, ba)$. Every $T \in \mathsf{Rec}(\mathbb{M}(\Sigma, I))$ (resp. $\Theta \in \mathsf{Rec}(\mathbb{R}(\Sigma, I))$) is recognized by a DACA [3] (resp. a DACMA). Via their global behaviors, asynchronous automata accept the corresponding trace-closed languages, and in particular, every regular trace-closed language (resp. trace-closed ω-regular language) is recognized by an I-diamond DFA (resp. I-diamond Muller automaton). In fact for every trace-closed $K \in \mathsf{REG}$, the minimal DFA $\mathfrak{A}_K$ accepting K is I-diamond.

3 From Regular Trace Languages to ω-Regular Trace Languages

We wish to extend the well-studied relations between regular and ω-regular languages to trace languages. We first look at reachability languages and their Boolean combinations, i.e. the weakly recognizable languages, and study how they can be obtained as a result of infinitary operations on regular trace languages. After this, we observe that the case of Büchi recognizability is not as straightforward and provide a resolution.

3.1 Infinitary Extensions of Regular Trace Languages

In the classification hierarchy of ω-regular languages, reachability and safety languages occupy the lowest levels. For trace languages we have the following.

Definition 3. *Let* $T \in \mathsf{Rec}(\mathbb{M}(\Sigma, I))$*. The* infinitary extension of T *is the* ω*-trace language given by* $\mathsf{ext}(T) := T \odot \mathbb{R}(\Sigma, I)$*.*

Extrapolating the definition of E-automata for word languages, we define E-automata for trace languages where a run is accepting if for each $a \in \Sigma$ some predefined local states from Q_a are reached. Formally, a *deterministic asynchronous E-automaton* (a *DAEA*) is a tuple $\mathfrak{a} = (\prod_{a\in\Sigma} Q_a, (\delta_a)_{a\in\Sigma}, q_0, \mathcal{F})$ with $\mathcal{F} \subseteq \prod_{a\in\Sigma} \mathscr{P}(Q_a)$. The DAEA $\mathfrak{a}$ accepts $\alpha \in \Sigma^\omega$ if for some $F = (F_a)_{a\in\Sigma} \in \mathcal{F}$ we have that $\mathsf{occ}_a(\rho_\alpha) \cap F_a \neq \emptyset$. Note that given a DACA $\mathfrak{A}$ with $L(\mathfrak{A}) = T$, in order to accept $\mathsf{ext}(T)$ any DAEA $\mathfrak{a}$ must infer the “global-state reachability” of $\mathfrak{A}$ by referring only to “local-state reachability” in $\mathfrak{a}$. A simple counterexample suffices to show that this is a difficult task.

Proposition 4. *There exist languages* $T \subseteq \mathsf{Rec}(\mathbb{M}(\Sigma, I))$ *such that* $\mathsf{ext}(T)$ *is not recognized by any DAEA.*

A similar argument can be drawn against a possible definition of deterministic asynchronous weak automata, defined in terms of SCCs that occur locally within Q_a for each $a \in \Sigma$. This means that the class of reachability languages resists characterization in terms of deterministic asynchronous cellular automata. We therefore concentrate on the classes of I-diamond automata and trace-closed reachability languages in the hope of finding reasonable characterizations.

First we note that the definition of infinitary extensions of a trace-closed languages is not sound with respect to trace equivalence of ω-words; i.e. if $T \in \mathsf{Rec}(\mathbb{M}(\Sigma, I))$ and $K = \Gamma^{-1}(T)$, then, in general, $\mathsf{ext}(K) \neq \Gamma^{-1}(\mathsf{ext}(T))$.

Example 5. Let $\Sigma = \{a, b, c\}$, and bIc. Define $K := [ab]_{\sim_I}$. Clearly K is trace-closed and, moreover, $acb \notin K$. Let $T = \Gamma(K)$. Clearly $abc^\omega, acbc^\omega, accbc^\omega, \ldots$ are equivalent words since they induce the same infinite trace which belongs to $\mathsf{ext}(T)$. However, while $abc^\omega \in \mathsf{ext}(K)$, $ac^+bc^\omega \nsubseteq \mathsf{ext}(K)$. ⊠

Definition 6. *Let* $K \subseteq \Sigma^*$ *be trace-closed. Define the* I-suffix extended *trace-closed language (or* I-suffix extension*) of* K *as* $K_I := K \cup \bigcup_{a\in\Sigma} [Ka^{-1}aI_a^*]_{\sim_I}$*.*

Due to the closure of $\mathsf{Rec}(\mathbb{M}(\Sigma, I))$ under concatenation and finite union [3], we know that K_I is regular whenever K is regular.

Proposition 7. *For a language* $T \in \mathsf{Rec}(\mathbb{M}(\Sigma, I))$*, let* $K = \Gamma^{-1}(T)$*, and let* K_I *be the* I*-suffix extension of* K*. Then it holds that* $\Gamma^{-1}(\mathsf{ext}(T)) = \mathsf{ext}(K_I)$*.*

Remark 8. In general $K_I \neq (K_I)_I$. However, iterated I-suffix extensions preserve the infinitary extension languages, i.e. $\mathsf{ext}(K) \subseteq \mathsf{ext}(K_I) = \mathsf{ext}((K_I)_I) = \mathsf{ext}(((K_I)_I)_I) \ldots$ and so on. ⊠

Proposition 7 provides us the basis for generating the class of weakly recognizable trace-closed languages corresponding to the recognizable subset of $\mathrm{BC}(\mathsf{ext}(\mathbb{M}(\Sigma, I)))$. Henceforth, whenever we speak of the language $\Gamma^{-1}(\mathsf{ext}(T))$ we refer to $\mathsf{ext}(\Gamma^{-1}(T)_I)$. Similarly, for a trace-closed language K we always mean $\mathsf{ext}(K_I)$ whenever we say $\mathsf{ext}(K)$.

Theorem 9. *A trace-closed language $L \subseteq \Sigma^\omega$ is recognized by an I-diamond DWA iff $L \in \mathrm{BC}(\mathsf{ext}(\mathcal{K}))$ for a set $\mathcal{K} \subseteq 2^{\Sigma^*}$ of trace-closed regular languages.*

3.2 Infinitary Limits of Regular Trace Languages

We now consider the *infinitary limit* operator. In the case of word languages, this operator extends regular languages to the family ω-regular languages that are DBA recognizable. This is not straight forward for traces, and here we seek an effective characterization of languages $T \in \mathsf{Rec}(\mathbb{M}(\Sigma, I))$, such that $\Gamma^{-1}(\mathsf{lim}(T))$ is recognized by an I-diamond DBA.

Definition 10 ([2]). *Let $T \in \mathsf{Rec}(\mathbb{M}(\Sigma, I))$, the* infinitary limit $\mathsf{lim}(T)$ *is the ω-trace language containing all $\theta \in \mathbb{R}(\Sigma, I)$ such that there exists a sequence $(t_i)_{i\in\mathbb{N}}, t_i \in T$ satisfying $t_i \sqsubset t_{i+1}$ and $\bigsqcup_{i\in\mathbb{N}} t_i = \theta$.*

It is open whether there exists any characterization for the class of languages recognized by the family of DACBAs, however there do exist regular languages $T \subseteq \mathbb{M}(\Sigma, I)$ such that $\mathsf{lim}(T)$ is not recognized by any DACBA [8]. In fact, even when relying on trace-closed word languages and I-diamond automata, we cannot hope to characterize these languages in the manner of infinitary extensions as demonstrated previously in Section 3.1.

Example 11. Let $\Sigma = \{a, b\}$, and aIb. Define $K := [(aa)^+(bb)^+]_{\sim_I}$ as the trace-closed language with even number of occurrences of a's and b's. The minimal DFA accepting this language is shown in Figure 2. If $T = \Gamma(K)$, then

$$\mathsf{lim}(T) = \Theta := \left\{ \theta \in \mathbb{R}(\Sigma, I) \;\middle|\; \begin{array}{l} |\theta|_a \text{ even}, |\theta|_b = \infty, \text{ or} \\ |\theta|_a = \infty, |\theta|_b \text{ even}, \text{ or} \\ |\theta|_a = |\theta|_b = \infty \end{array} \right\}$$

The trace-closed language $L = \Gamma^{-1}(\Theta)$ consists of all infinite words $\alpha \in \Sigma^\omega$ that satisfy the same conditions as $\theta \in \Theta$ above. ⊠

It is easy to verify that the DFA of Figure 2 does not accept L when equipped with a Büchi acceptance condition. For instance, the automaton can loop forever in states 4, 6, and 7, thereby witnessing infinitely many a's and b's, without ever visiting state 8.

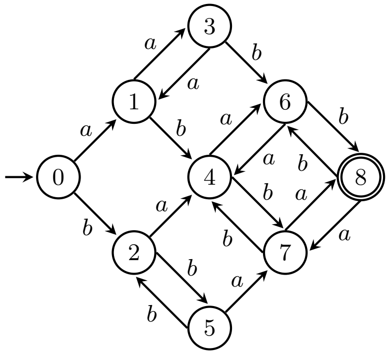

Fig. 2. The minimal DFA recognizing language K of Example 11

Proposition 12. *There exists no I-diamond deterministic parity automaton, and therefore no I-diamond deterministic Büchi automaton, that recognizes the language $L \subseteq \Sigma^\omega$ as described in Example 11.*

Corollary 13. *There exists a family $\mathcal{K}$ of regular trace-closed languages, namely $\mathcal{K} := \{[(a^m)^+(b^n)^+]_{\sim_I} \mid m,n \geq 2\}$ over $\Sigma = \{a,b\}$, such that given $T = \Gamma(K)$ for any $K \in \mathcal{K}$, there exists no I-diamond DBA recognizing $\Gamma^{-1}(\mathsf{lim}(T))$.*

Definition 14. *A trace-closed language $K \subseteq \Sigma^*$ is I-*limit-stable *(or simply* limit-stable*) if $\mathsf{lim}(K)$ is also trace-closed. By extension, $T \subseteq \mathbb{M}(\Sigma, I)$ is* limit-stable *if $\Gamma^{-1}(T)$ is.*

Given an automaton and its states p, q, we write $p \xrightarrow{u} q$ if some $u \in \Sigma^*$ leads from p to q, and $p \overset{u}{\Rightarrow} q$ if a final state is also visited.

Definition 15. *Given (Σ, I), let $\mathfrak{A} = (Q, \Sigma, q_0, \delta, F)$ be an I-diamond automaton. $\mathfrak{A}$ is F, I-*cycle closed*, if for all $u \sim_I v$ and all q we have $q \overset{u}{\Rightarrow} q$ iff $q \overset{v}{\Rightarrow} q$.*

We can now give an effective characterization of limit-stable languages.

Theorem 16. *For any $T \in \mathsf{Rec}(\mathbb{M}(\Sigma, I))$ and $K = \Gamma^{-1}(T)$, the following are equivalent:*

(a) K, and therefore T, is limit-stable.
(b) For all sequences $(t_i)_i = t_0 \sqsubset t_1 \sqsubset t_2 \cdots \subseteq T$ and all sequences $(u_i)_i$ with $u_i \in \Gamma^{-1}(t_i)$, there exists a subsequence $(u_{j_i})_i$ and a sequence $(v_{j_i})_i$ of proper prefixes $v_{j_i} \sqsubset u_{j_i}$ with $|v_{j_i}| < |v_{j_{i+1}}|$ and $v_{j_i} \in K$ for all $i \in \mathbb{N}$.
(c) Any DFA $\mathfrak{A}$ recognizing K is F, I-cycle closed.

Proof. (a) $\Longrightarrow$ (b): If (b) is false, then we may choose a sequence $(t_i)_i$ of traces in T with the property that for some sequence $(u_i)_i$ of linearizations of $(t_i)_i$, every subsequence $(u_{n_i})_i$, and every sequence $(v_{n_i})_i$ of proper prefixes $v_{n_i} \sqsubset u_{n_i}$, $v_{n_i} \in K$, we have $\sup_i |v_{n_i}| < \infty$. Since $|\Sigma| < \infty$ we have that Σ^∞ is a compact space. Hence $(u_i)_i$ has a converging subsequence $(u_{m_i})_i$. Because every subsequence of $(u_i)_i$ has the properties given in the previous sentence, so does $(u_{m_i})_i$. Let $\alpha = \lim_{i\to\infty} u_{m_i}$. Then $\alpha \sim_I \beta$ for some $\beta = x \cdot y_1 \cdot y_2 \cdots$ with $x \cdot y_1 \cdots y_i \in \Gamma^{-1}(t_{m_i})$. Hence, $\beta \in \mathsf{lim}(L)$. But, by construction, $\alpha \notin \mathsf{lim}(K)$ because for some $n \in \mathbb{N}$ no prefix of length $> n$ of α is in K.

(b) $\Longrightarrow$ (a): Let $\theta = \bigsqcup_i t_i$ for traces $t_i \in T$. We may assume that $t_i \sqsubset t \sqsubset t_{i+1}$ implies $t \notin T$. Let $\alpha \in \Gamma^{-1}(\theta)$. Then we pick prefixes $(w_i)_i$ of α, such that w_i is of minimal length with $t_i \sqsubseteq \Gamma(w_i)$. Consider the subsequence $(t_{2i})_i$ of $(t_i)_i$. Each w_{2i+1} is a prefix of some linearization of $t_{2(i+1)}$, say $u_{2(i+1)}$. We apply (b) to the sequence $(t_{2i})_i$ and get a sequence $(v_{2i})_i$ of proper prefixes of the u_{2i}, such that $\sup_i |v_{2i}| = \infty$ and $v_{2i} \in K$. We now have to show that v_{2i} is already a prefix of w_{2i-1}. Suppose not, i.e. $w_{2i-1} \sqsubset v_{2i} \sqsubset u_{2i}$. Then this would give a trace $t \in T$ with $t_{2i-1} \sqsubset t \sqsubset t_{2i}$.

(a) $\Longrightarrow$ (c): Suppose $\mathfrak{A}$ is not F,I-cycle closed. Then there exists $q \in Q$ and $u \sim_I v$ with $q \overset{u}{\Rightarrow} q$ but not $q \overset{v}{\Rightarrow} q$. Since $\mathfrak{A}$ is I-diamond, this means that the run $q \xrightarrow{v} q$ exists, but does not visit a final state. Now pick $x \in \Sigma^*$ with $q_0 \xrightarrow{x} q$. Then $\alpha = x \cdot u^\omega \in \mathsf{lim}(K)$ and $\beta = x \cdot v^\omega \notin \mathsf{lim}(L)$. But clearly $\alpha \sim_I \beta$ implies that $\mathsf{lim}(K)$ is not trace-closed.

(c) $\Longrightarrow$ (a): Let $\alpha \sim_I \beta$ and let $\alpha \in \mathsf{lim}(K)$. Take $\mathfrak{A} = \mathfrak{A}_K$ and consider extended transition profiles $\tau_w \subseteq Q \times \{0,1\} \times Q$ for $w \in \Sigma^*$ defined by $(p,1,q) \in \tau_w$ iff $p \overset{w}{\Rightarrow} q$ and $(p,0,q) \in \tau_w$ iff $p \xrightarrow{w} q$ but not $p \overset{w}{\Rightarrow} q$. Then we can factorize $\alpha = uv_0v_1v_2\cdots$ for finite words $u, v_0, v_1, \ldots$ with $\tau_u \cdot \tau_{v_i} = \tau_u$ and $\tau_{v_i} \cdot \tau_{v_{i+1}} = \tau_{v_i}$. Likewise we can factorize $\beta = u'v'_0v'_1\cdots$.

Next, we observe that we find $r \in \mathbb{N}$ with $\Gamma(u'v'_0) \sqsubseteq \Gamma(uv_0\cdots v_r)$. This gives $x \in \Sigma^*$ with $u'v'_0 \cdot x \sim_I uv_0\cdots v_r$. Conversely, there exists $m \in \mathbb{N}$ with $\Gamma(uv_0\cdots v_{r+1}) \sqsubseteq \Gamma(u'v'_0\cdots v'_m)$ and therefore $y \in \Sigma^*$ with $u'v'_0\cdots v'_m \sim_I uv_0\cdots v_rv_{r+1}y \sim_I u'v'_0xv_{r+1}y$, which implies $xv_{r+1}y \sim_I v'_1\cdots v'_m$.

Notice that if $q_0 \xrightarrow{u} q$ and $q_0 \xrightarrow{u'} q'$, then (by trace equivalence and the fact that $\mathfrak{A}$ is I-diamond) we have $q' \xrightarrow{x} q$. Likewise we have $q \xrightarrow{y} q'$ and $q' \xrightarrow{xv_{r+1}y} q'$. Now we can apply (c) to see that $q' \overset{xv_{r+1}y}{\Longrightarrow} q'$ iff $q' \overset{v'_1\cdots v'_m}{\Longrightarrow} q'$. However, since $\alpha \in \mathsf{lim}(K)$, since $\tau_{v_{r+1}} = \tau_{v_i}$ for all i, and since $q \overset{v_{r+1}}{\Longrightarrow} q$, we have $q' \overset{xv_{r+1}y}{\Longrightarrow} q'$. Hence, $q' \overset{v'_1\cdots v'_m}{\Longrightarrow} q'$. Since furthermore $\tau_{v'_1\cdots v'_m} = \tau_{v'_i}$, we have for all $i, q' \underset{F}{\overset{v'_i}{\Rightarrow}} q'$ whence $\beta \in \mathsf{lim}(K)$. ■

Corollary 17. *Let $K = \Gamma^{-1}(T)$ for some $T \in \mathsf{Rec}(\mathbb{M}(\Sigma, I))$. Given the automaton $\mathfrak{A}_K$, it is decidable in time $\mathcal{O}(|Q|^2 \cdot |\Sigma|(|\Sigma| + \log|Q|))$ whether or not K is limit-stable.*

3.3 Characterization of Regular Infinitary Trace-Closed Languages

In [2], it was shown that for every recognizable ω-trace language $\Theta \subseteq \mathbb{R}(\Sigma, I)$ the corresponding ω-regular trace-closed language $L = \Gamma^{-1}(\Theta)$ is recognized by an I-diamond deterministic Muller automaton (DMA). On the other hand, it is not the case that every I-diamond DMA recognizes a trace-closed language. Similar to the property of F, I-cycle closure for DBAs, we present a condition over the acceptance component $\mathcal{F}$ of I-diamond DMAs to enable a characterization.

Given an automaton, two of its states p, q, and a word $u \in \Sigma^*$, we denote with $\mathsf{occ}(p \xrightarrow{u} q)$ the set of states occurring in the run from p to q over u.

Definition 18. *Given (Σ, I), an I-diamond DMA $\mathfrak{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ is said to be $\mathcal{F}, I$-*cycle closed *if for all $u, v \in \Sigma^*$ such that $u \sim_I v$, and all $q \in Q$, we have $\mathsf{occ}(q \xrightarrow{u} q) \in \mathcal{F}$ iff $\mathsf{occ}(q \xrightarrow{v} q) \in \mathcal{F}$.*

$\mathcal{F}, I$-cycle closure was mentioned in [8, Chapter 7] under a different term. We obtain an independent proof of the following result by using an approach very similar to that we used to show the equivalence Theorem 16:(a) $\Leftrightarrow$ (c).

Theorem 19 (also cf. [8]). *For any language $\Theta \subseteq \mathbb{R}(\Sigma, I)$ of infinite traces, Θ is recognized by a DACMA if and only if the trace-closed language $L = \Gamma^{-1}(\Theta)$ is recognized by an $\mathcal{F}, I$-cycle closed DMA.*

4 A Borel-Like Classification

Any I-diamond DWA recognizing a trace-closed language is trivially F, I-cycle closed since for any word $u \in \Sigma^*$ and any $q \in Q$, it holds that $q \overset{u}{\Rightarrow} q$ if and if all states in the path taken by u are accepting. This is because a path from q back to itself also implies an SCC, and therefore any $v \sim_I u$ will also remain in the same SCC which comprises solely of accepting states.

It is also straightforward that for a limit-stable language K, the complement language $\overline{\mathsf{lim}(K)}$ of K's infinitary extension is also recognized by an F, I-cycle closed deterministic co-Büchi automaton (DcBA). The following result is then a consequence of Theorem 9, Theorem 16, and the definitions.

Theorem 20. *A trace-closed language $L \subseteq \Sigma^\omega$ is recognized by an I-diamond DWA if and only if it is recognized by both an F, I-cycle closed DBA and an F, I-cycle closed DcBA.*

This result is in nice correspondence with the classical Borel level where weakly recognizable languages are precisely those that lie in the intersection of deterministic Büchi and deterministic co-Büchi recognizable languages. Finally, we now demonstrate that the class of limit-stable languages is expressive enough to generate all ω-regular trace-closed languages.

In [2], it was shown using algebraic arguments that every recognizable ω-trace language can be expressed as a finite Boolean combination of "restricted" lim-languages. This result also extends to the corresponding trace-closed linearization languages. Our characterization of limits of limit-stable languages allows for a first automata-theoretic equivalence result.

Theorem 21. *Let L be a trace-closed ω-language. L is ω-regular iff L is a finite Boolean combination of infinitary limits of limit-stable languages, i.e. a finite Boolean combination of F, I-cycle closed DBA recognizable languages.*

Proof. Recall the definition of DACMAs (cf. Sec. 2), and the result that every recognizable ω-trace language is recognized by a DACMA [2].

Let $L \subseteq \Sigma^\omega$ be recognizable, trace-closed. Pick a DACMA $\mathfrak{a}$ recognizing L. Recall that the global transition behavior of $\mathfrak{a}$ gives an I-diamond DFA, and we denote this DFA by $\mathfrak{A} = (\prod_{a\in\Sigma} Q_a, \Sigma, q_0, \delta)$. Given $q \in Q_a$ we obtain a DBA $\mathfrak{A}_q = (\prod_{a\in\Sigma} Q_a, \Sigma, q_0, \delta, F_q)$, where $F_q = \{q\} \times \prod_{b\neq a} Q_b$. Note that $\mathfrak{A}_q$ is F_q, I-cycle closed, because for any $q' \in \prod_{a\in\Sigma} Q_a$ and all $u \sim_I v$ with $q' \xrightarrow{u} q'$ and $q' \xrightarrow{v} q'$ we have[2] $\mathsf{occ}_a(q' \xrightarrow{u} q') = \mathsf{occ}_a(q' \xrightarrow{v} q')$. Now:

$$L = \bigcup_{(F_a)_{a\in\Sigma}\in\mathcal{F}} \bigcap_{a\in\Sigma} \bigcap_{q\in F_a} L(\mathfrak{A}_q) \cap \bigcap_{q\notin F_a} \overline{L(\mathfrak{A}_q)}$$

■

We therefore obtain a Borel-like classification for recognizable ω-trace languages where the lowest level is occupied by reachability and safety languages. At the next level, we have infinitary limits of limit-stable languages and their complements. And the Boolean combinations of these languages generate the class of all recognizable ω-trace languages.

5 Conclusion

The infinitary extension operator ext and the infinitary limit operator lim offer natural mechanisms for obtaining ω-languages by expressing reachability and liveness conditions over regular languages. While in the case of word languages, these ω-languages have well-known characterizations in terms of specific classes of ω-automata, it is not easy to generalize these observations to trace languages. Analogous characterizations of recognizable ω-trace languages in terms of classes of deterministic asynchronous automata either do not exist (reachability conditions) or impose a high technical complexity (liveness conditions).

The results of this paper demonstrate that a classification of recognizable ω-trace languages in terms of trace-closed word languages is both meaningful and efficient. Once in the realm of words, for any trace language $T \in \mathsf{Rec}(\mathbb{M}(\Sigma, I))$ we investigated the relationship between its infinitary extension $\mathsf{ext}(T)$ and the infinitary extension $\mathsf{ext}(K)$, where $K = \Gamma^{-1}(T)$. We showed that any such K can be modified to K_I such that $\mathsf{ext}(K_I)$ is also trace-closed and thus corresponds to the linearizations of $\mathsf{ext}(T)$. We also showed that Boolean combinations of trace-closed languages $\Gamma^{-1}(\mathsf{ext}(T)), T \in \mathsf{Rec}(\mathbb{M}(\Sigma, I))$, are precisely the languages recognized by the class of I-diamond DWAs. In a similar vein, a trace-closed language $K \in \mathsf{Rec}(\Sigma^*)$ is limit-stable precisely when $\mathsf{lim}(K)$ is also trace-closed and recognized by F, I-cycle closed DBA, which can be obtained from any DFA recognizing K. Moreover, we showed that it is efficiently decidable whether or not a trace-closed word language K, and therefore $T = \Gamma(K)$, is limit-stable.

[2] This can be proven by an induction on the number of swapping operations needed to obtain v from u.

It must be noted that if the independence relation I over the alphabet is empty, then we obtain the well-known theorems for ω-regular word languages as special cases of our results. In this manner, our characterizations have two interesting consequences. First, that I-diamond DWA recognizable languages are precisely those that are both I-diamond det. Büchi and I-diamond det. co-Büchi recognizable. Second, that every recognizable language of infinite traces can be expressed as a Boolean combination of languages $\mathsf{lim}(T)$ for limit-stable languages T. This, in turn, gives rise to a Borel-like classification hierarchy for trace languages in terms of trace-closed word languages.

As a next step, we would like to investigate whether these classes of languages can also be characterized in terms of logic. Such a characterization will allow for a direct comparison with Borel levels. Also, in the manner of Landweber's result for ω-regular word languages in general, we would like to have the ability to decide whether or not a given trace-closed word language is recognized by an F, I-cycle closed deterministic Büchi automaton.

Acknowledgement. We would like to thank Wolfgang Thomas and Christof Löding for encouragement and numerous fruitful discussions.

References

1. Chaturvedi, N.: Toward a Structure Theory of Regular Infinitary Trace Languages. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014, Part II. LNCS, vol. 8573, pp. 134–145. Springer, Heidelberg (2014)
2. Diekert, V., Muscholl, A.: Deterministic Asynchronous Automata for Infinite Traces. Acta Informatica 31(4), 379–397 (1994)
3. Diekert, V., Rozenberg, G. (eds.): The Book of Traces. World Scientific (1995)
4. Gastin, P., Petit, A.: Asynchronous Cellular Automata for Infinite Traces. In: Kuich, W. (ed.) ICALP 1992. LNCS, vol. 623, pp. 583–594. Springer, Heidelberg (1992)
5. Löding, C.: Efficient minimization of deterministic weak ω-automata. Information Processing Letters 79(3), 105–109 (2001)
6. Madhavan, M.: Automata on Distributed Alphabets. In: D'Souza, D., Shankar, P. (eds.) Modern Applications of Automata Theory. IISc Research Monographs Series, vol. 2, pp. 257–288. World Scientific (May 2012)
7. Mazurkiewicz, A.: Trace Theory. In: Brauer, W., Reisig, W., Rozenberg, G. (eds.) Petri Nets: Applications and Relationships to Other Models of Concurrency. LNCS, vol. 255, pp. 278–324. Springer, Heidelberg (1987)
8. Muscholl, A.: Über die Erkennbarkeit unendlicher Spuren. PhD thesis (1994)
9. Perrinand., D., Pin, J.-É.: Automata and Infinite Words. In: Infinite Words: Automata, Semigroups, Logic and Games. Pure and Applied Mathematics, vol. 141. Elsevier (2004)
10. Staiger, L.: Subspaces of $GF(q)^\omega$ and Convolutional Codes. Information and Control 59(1-3), 148–183 (1983)
11. Zielonka, W.: Notes on Finite Asynchronous Automata. R.A.I.R.O. – Informatique Théorique et Applications 21, 99–135 (1987)