Logical Definability on Infinite Traces *

Werner Ebinger and Anca Muscholl

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

Abstract

The main results of the present paper are the equivalence of definability by monadic second-order logic and recognizability for real trace languages, and that first-order definable, star-free, and aperiodic real trace languages form the same class of languages. This generalizes results on infinite words and on finite traces to infinite traces. It closes an important gap in the different characterizations of recognizable languages of infinite traces.

1 Introduction

In the late 70's, A. Mazurkiewicz introduced the notion of trace as a suitable mathematical model for concurrent systems [16] (for surveys on this topic see also [1, 6, 10, 17]). In this framework, a concurrent system is seen as a set Σ of atomic actions together with a fixed irreflexive and symmetric independence relation $I \subseteq \Sigma \times \Sigma$. The relation I specifies pairs of actions which can be carried out in parallel. It generates an equivalence relation on the set of sequential observations of the system. As this relation is actually a congruence relation, it defines a quotient monoid of Σ^*, called *trace monoid*. These monoids are also called *free partially commutative monoids* and have been first studied in combinatorics by Cartier and Foata [5].

Actions in a sequential system are linearly ordered, whereas a concurrent run (trace) corresponds to a restricted Σ-labelled partial order. Traces can also be viewed as dependence graphs, i. e. as labelled, acyclic graphs, where vertices are labelled with actions and edges exist precisely between vertices with dependent labels.

A natural framework for studying non-terminating concurrent systems (e. g. operating systems, transaction systems) is provided by extending Mazurkiewicz

*This research has been supported by the EBRA working group No. 6317 ASMICS II.

traces to infinite traces. Infinite traces are given immediately by considering the extension to infinite dependence graphs. In this paper we are concerned only with dependence graphs where every vertex has a finite past, i. e. every action can be performed within a finite delay. Infinite traces corresponding to this kind of dependence graphs are called *real traces*. Alternatively, real traces can be viewed as the ideal completion (with respect to the prefix order) of the monoid of finite traces [17].

The family of recognizable real trace languages has been first investigated by P. Gastin [12] from the viewpoint of recognition by saturating homomorphisms. Consequently, the extension of Ochmański's theorem from finitary trace languages to infinitary trace languages provided a characterization by concurrent-rational expressions [14]. Concerning recognition by means of finite state automata, a suitable model for trace languages is given by automata with distributed control, namely asynchronous (cellular) automata [26, 27]. With appropriate extensions of classical acceptance conditions (Büchi, Muller), it has been shown that the class of recognizable real trace languages corresponds to the family of languages accepted by non-deterministic Büchi [13], respectively deterministic Muller asynchronous (cellular) automata [9]. Solely the logical characterization of recognizability in terms of monadic second-order formulae remained open so far and is provided by this paper.

Motivated by applications in the field of verification and specification of distributed systems (e. g. model checking), the question of determining the expressive power of logic in the context of infinite traces is of particular interest. In the present paper we provide answers to some of the remaining open problems in the theory of real traces [8, for some recent open problems]. We show for example that monadic second-order logic corresponds to recognizability, thus being decidable. Our approach is independent of any model of trace automata.

In the weaker first-order logic framework, we give again a proper generalization from the theory of (ω-) words and finite traces and show that the family of star-free (aperiodic, respectively) sets of real traces coincides with the family of first-order definable languages.

The paper is organized as follows: In Section 2 we recall some basic notions of trace theory, together with some properties of recognizable sets. In Section 3 we show the equivalence of monadic second-order logic and recognizability for real trace languages, extending the characterization for languages of finite traces [25]. Together with a result obtained in Section 4 this provides a new proof for Métivier's [18, Theorem 2.3] and Ochmański's [19, Lemma 8.2] result on the recognizability of the Kleene-iteration of connected recognizable languages of finite traces. In Section 4, we consider first-order logic on real traces. First we provide direct transformations between formulae interpreted on finite words and formulae interpreted on finite traces. Then we show that first-order definable languages are exactly the star-free languages. Finally, we show the equivalence of star-freeness and aperiodicity extending characterizations obtained for languages of finite traces [15]. Some of our ideas have been proposed independently by H. J. Hoogeboom, W. Thomas, and W. Zielonka (personal communication).

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.* The notations $D(a) = \{ b \in \Sigma \mid (a, b) \in D \}$ and $D(\Sigma') = \bigcup_{a \in \Sigma'} D(a)$, $\Sigma' \subseteq \Sigma$, will be used throughout the paper.

The monoid of *finite traces,* $\mathbb{M}(\Sigma, D)$, is defined as a quotient monoid with respect to the congruence relation generated by the independence relation I, i. e. $\mathbb{M}(\Sigma, D) = \Sigma^*/\{ ab \equiv ba \mid (a, b) \in I \}$. The empty trace (and the empty word as well) will be denoted by 1. A trace can be identified with its *dependence graph,* i. e. with (an isomorphism class of) a labelled, acyclic, directed graph $[V, E, \ell]$, where V is a set of vertices labelled by $\ell : V \to \Sigma$ and E is a set of edges between vertices with dependent labels. More precisely, we have for every $x, y \in V$, $(\ell(x), \ell(y)) \in D$ if and only if $x = y$ or $(x, y) \in E$ or $(y, x) \in E$. Thus, we associate to every word $a_1 \cdots a_n$, $a_i \in \Sigma$, the vertex set $V = \{1, \ldots, n\}$ labelled as $\ell(i) = a_i$, together with the edge set $E = \{(i, j) \mid 1 \leq i < j \leq n,\ (a_i, a_j) \in D\}$. This notion provides a natural definition of infinite traces by means of infinite dependence graphs. We denote by $\mathbb{G}(\Sigma, D)$ the set of dependence graphs with a countable set of vertices V, such that $\ell^{-1}(a)$ is well-ordered for every $a \in \Sigma$.

$\mathbb{G}(\Sigma, D)$ is a monoid with respect to the concatenation $[V_1, E_1, \ell_1][V_2, E_2, \ell_2] = [V, E, \ell]$, where $[V, E, \ell]$ is the disjoint union of $[V_1, E_1, \ell_1]$ and $[V_2, E_2, \ell_2]$, together with additional edges $(v_1, v_2) \in V_1 \times V_2$, whenever $(\ell_1(v_1), \ell_2(v_2)) \in D$ holds. The identity is the empty graph $1 = [\emptyset, \emptyset, \emptyset]$. The concatenation is immediately extendable to finite and infinite products. Let $(g_n)_{n \geq 0} \subseteq \mathbb{G}(\Sigma, D)$. The infinite product $g = g_0 g_1 \ldots \in \mathbb{G}(\Sigma, D)$ is the disjoint union of the g_n, together with additional edges from g_n to g_m for $n < m$ between vertices with dependent labels. Thus, we define the *ω-iteration* of $A \subseteq \mathbb{G}(\Sigma, D)$ as $A^\omega = \{ g_0 g_1 \ldots \mid g_n \in A,\ \forall n \geq 0 \}$ (note that for $1 \in A$ we have $A^\omega = A^* \cup (A \setminus 1)^\omega$).

We denote by Σ^ω the set of infinite words over the alphabet Σ (i. e. mappings from $\mathbb{N}$ to Σ), and by Σ^∞ the set of finite and infinite words $\Sigma^* \cup \Sigma^\omega$. The *canonical mapping* $\varphi : \Sigma^* \to \mathbb{M}(\Sigma, D)$ associating to a sequence its trace (dependence graph) can be naturally extended to Σ^∞, i. e. $\varphi : \Sigma^\infty \to \mathbb{G}(\Sigma, D)$. The image $\varphi(\Sigma^\infty) \subseteq \mathbb{G}(\Sigma, D)$ is called the set of *real traces* and is denoted by $\mathbb{R}(\Sigma, D)$. Real traces correspond to (in)finite graphs, where every vertex has finitely many predecessors. A word $w \in \Sigma^\infty$ is called a representative of $t \in \mathbb{R}(\Sigma, D)$ if $\varphi(w) = t$. Throughout this paper we abbreviate $\mathbb{R}(\Sigma, D)$ ($\mathbb{M}(\Sigma, D)$, respectively) by $\mathbb{R}$

($\mathbb{M}$, respectively).

Observe that $\mathbb{R}$ is not a submonoid of $\mathbb{G}(\Sigma, D)$, as e.g. $a^\omega a$ is not a real trace anymore. A semantically satisfactory definition of the concatenation operation is given by extending the theory to complex traces [7]. Since we consider real traces only, we have chosen here the approach of viewing the concatenation as a partially defined operation on $\mathbb{R}$, i. e., $t = t_1 t_2$ for $t_1, t_2 \in \mathbb{R}$ is defined only if $t \in \mathbb{R}$. Note that this condition is always satisfied if t_1 is a finite trace.

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 $\varphi : \Sigma^\infty \to \mathbb{R}$.

We denote by $\mathrm{alph}(t)$ the set of letters occurring in a trace t. We also use the abbreviation $(t, u) \in I$ for $\mathrm{alph}(t) \times \mathrm{alph}(u) \subseteq I$.

A trace is called *connected* if its dependence graph is connected. A language is called *connected* if all its elements are connected. Every trace $t \in \mathbb{R}$ can be decomposed into connected components $t = t_1 \dot\cup \ldots \dot\cup t_n$, i. e. every t_i is a connected factor of $t = t_1 \cdots t_n$ and $(t_i, t_j) \in I$, for $1 \leq i \neq j \leq n$. Let $A \subseteq \mathbb{M}$, then the language of its connected components is defined as $\mathrm{CC}(A) = \{\, u \in \mathbb{M} \mid u$ is a connected component of some $t \in A \,\}$.

The set of letters occurring infinitely often in a real trace t is denoted $\mathrm{alphinf}(t)$.

We conclude this section with an example. Consider two concurrent processes P_1, P_2, given by the instruction sequences

$$P_i : \ \textbf{while} \text{ true } \textbf{do } x_i := f_i(x_i, y);\ y := i \ \textbf{endwhile} \ .$$

We have four (atomic) instructions $a = (x_1 := f_1(x_1, y))$, $b = (y := 1)$, $c = (x_2 := f_2(x_2, y))$ and $d = (y := 2)$, with the independence relation $I = \{(a, c), (c, a)\}$. Sequences of instructions satisfying that no process is idle forever are given e.g. by the real trace language $\{t \in \{ab, cd\}^\omega \mid \{b, d\} \subseteq \mathrm{alphinf}(t)\}$.

2.2 Recognizable Infinitary Word and Trace Languages

In this section we recall some properties of the family of *recognizable* subsets of Σ^∞ and $\mathbb{R}$, denoted by $\mathrm{Rec}(\Sigma^\infty)$ and $\mathrm{Rec}(\mathbb{R})$, respectively. Recognizable infinitary word languages can be characterized in several ways. The most familiar one involves finite-state *automata,* equipped with suitable acceptance conditions. These conditions specify (possibly partially) the set of states which have to occur infinitely often on an accepting path. A further characterization is given by *ω-rational expressions,* which are formed over finite languages of finite words by using the operations union, concatenation, Kleene-star and ω-iteration. Following the definition of the infinite product for dependence graphs, we have for $A \subseteq \Sigma^*$, $A^\omega = \{\, w \in \Sigma^\infty \mid w = w_0 w_1 \ldots,$ with $w_n \in A$ for $n \geq 0 \,\}$. In particular, if the empty word belongs to A, then $A^\omega = A^* \cup (A \setminus \{1\})^\omega$.

Finally, from the logical viewpoint, recognizability of (infinitary) word languages corresponds to definability in the *monadic second-order logic* framework studied by Büchi [4].

One possible way to define *recognizable real trace languages* is by *saturating homomorphisms* [12]. Let $\eta : \mathbb{M} \to S$ be a homomorphism to a finite monoid S. A real trace language $A \subseteq \mathbb{R}$ is *recognized* by η if for any sequence $(t_n)_{n\geq 0} \subseteq \mathbb{M}$ the following saturation property holds:

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

The saturation property leads, together with a standard Ramsey argument, to a representation of A as $A = \bigcup_{(s,e)\in P_A} \eta^{-1}(s)\, \eta^{-1}(e)^\omega$ with $P_A = \{\, (s,e) \in S^2 \mid se = s, e^2 = e \text{ and } \eta^{-1}(s)\, \eta^{-1}(e)^\omega \cap A \neq \emptyset \,\}$. An equivalent definition uses Arnold's *syntactic congruence* [3]. For $A \subseteq \mathbb{R}$, two finite traces $u, v \in \mathbb{M}$ are congruent if and only if :

$$\forall x, y \in \mathbb{M} : \quad x(uy)^\omega \in A \ \Leftrightarrow\ x(vy)^\omega \in A \ ,$$
$$\forall x, y, z \in \mathbb{M} : \quad xuyz^\omega \in A \ \Leftrightarrow\ xvyz^\omega \in A \ .$$

We denote the syntactic congruence by $\equiv_A$ and consider the canonical homomorphism $\eta : \mathbb{M} \to \mathrm{Synt}(A)$, where $\mathrm{Synt}(A) = \mathbb{M}/\equiv_A$ is the *syntactic monoid* of A. Then $A \in \mathrm{Rec}(\mathbb{R})$ if and only if the syntactic congruence $\equiv_A$ has finite index and $\eta : \mathbb{M} \to \mathrm{Synt}(A)$ recognizes A. Furthermore, for $A \subseteq \mathbb{R}$ we have $\mathrm{Synt}(A) = \mathrm{Synt}(\varphi^{-1}(A))$ [12].

Due to the partial commutativity there can be no equivalence between recognizability and rational expressions with the Kleene-star as iteration operator. The solution to this problem was given by E. Ochmański [19], who introduced the concept of concurrent iteration. With this notion the family of recognizable finitary trace languages shows to coincide with the family of *c-rational* (or co-rational) languages. In the infinitary case, ω-rational expressions (formed by using in addition the ω-iteration) precisely characterize recognizable ω-word languages. Again, the result has its counterpart for real trace languages, by using the concurrent ω-iteration instead of the usual ω-iteration [14]. For further details on c-rational real trace languages, we refer to Section 3.1.

3 Monadic Second-Order Logic over Real Traces

In order to specify properties of real trace languages by logical formulae, a real trace $t \in \mathbb{R}$ will be identified with its dependence graph. Logical *formulae* are defined over *structures* of the form $(V, <, (P_a)_{a\in\Sigma})$, corresponding to dependence graphs $[V, E, \ell]$, where $<$ is the partial order induced by E and $P_a = \{v \in V \mid \ell(v) = a\}$, $a \in \Sigma$ (recall that the restriction of $<$ to P_a is a well-founded total order). We allow the empty structure ($V = \emptyset$) in order to include the empty trace. We use first-order variables x, y, z, $\ldots$ ranging over the vertex set V and set variables X, Y, Z, $\ldots$ ranging over sets of vertices. Formulae are defined inductively as follows.

- *Atomic formulae* are given by the first-order predicates $x < y$, $P_a(x)$ and the (monadic) second-order predicate $x \in X$, with x, y, X denoting variables (X is a set variable) and $a \in \Sigma$.

- *Logical connectives:* If ψ_1 and ψ_2 are formulae, then $(\psi_1 \vee \psi_2)$ and $(\neg\psi_1)$ are formulae, too.

- *Quantifiers:* If ψ is a formula, x is a first-order variable, and X is a monadic second-order variable, then $\exists x\psi$ and $\exists X\psi$ are formulae, too.

We refer to the monadic second-order logic system introduced above as MSO. We freely use also $\wedge$, $\rightarrow$, $\forall$ and abbreviations like $X \subseteq Y$ for $\forall x\ (x \in X \rightarrow x \in Y)$ and $x = y$ for $\neg(x < y) \wedge \neg(y < x) \wedge (\bigvee_{a\in\Sigma}(P_a(x) \wedge P_a(y)))$. A real trace t is a model for a sentence ψ (i. e. a formula without free variables), if ψ is satisfied by t under the canonical interpretation (in symbols $t \models \psi$). This means that variables are interpreted by vertices, respectively sets of vertices of the dependence graph $G(t)$ of t, the predicate $P_a(x)$ is "x is labelled with $a \in \Sigma$", the predicate $<$ is interpreted as the partial order of $G(t)$, and $y \in X$ is "y belongs to X". The real trace language defined by a MSO sentence ψ is given by $\{t \in \mathbb{R} \mid t \models \psi\}$. The logical framework introduced above will also be used for defining languages of finite and infinite words. The difference consists in the interpretation of $<$ as a total order in models corresponding to words.

Example 1 The property "ab is a (trace) factor" can be expressed by the sentence

$$\exists x\, \exists y\, (P_a(x) \wedge P_b(y) \wedge \neg(y < x) \wedge \neg\, \exists z\, (x < z \wedge z < y))\ .$$

Note that the expressive power of monadic second-order logic with respect to trace properties does not depend on whether the partial order relation $<$ or the edge relation E_H of the Hasse diagram of a trace (or the edge relation of the dependence graph) are chosen to be part of the logical framework, since one can be expressed into the other. The edge relation E_H generalizes the successor relation on words and is expressible by the partial order (even in first-order logic):

$$x\ E_H\ y \quad \text{iff} \quad x < y \wedge \neg\, \exists z\, (x < z \wedge z < y)\ ,$$

and conversely (in second-order logic):

$$x < y \quad \text{iff} \quad \exists x' \Big(x\ E_H\ x' \wedge \forall X (x' \in X \wedge \forall z \forall z' (z \in X \wedge z\ E_H\ z' \rightarrow z' \in X) \rightarrow y \in X) \Big)\ .$$

Therefore we are free to use both, the partial order $<$ and the Hasse diagram edge relation, in monadic second-order formulae.

3.1 Equivalence of Recognizability and Monadic Second-Order Logic

The aim of this section is to show the equivalence of recognizability and definability in monadic second-order logic for real trace languages. One possible way to obtain this result is by using automata-theoretic characterizations. We use here a different approach, which turns out to be more elegant. Our proof is based on the characterization of recognizability by *c-rational languages* [19, 14], which we describe in the following. First, let us define for $A \subseteq \mathbb{M}$ the *concurrent iteration* $A^{c*} \subseteq \mathbb{M}$, respectively *concurrent ω-iteration* $A^{c\omega} \subseteq \mathbb{R}$ by $A^{c*} := (\mathrm{CC}(A))^*$, respectively $A^{c\omega} := (\mathrm{CC}(A))^\omega$ (recall the notation $\mathrm{CC}(A)$ for the connected components of elements of A).

Now, c-rational trace languages form the least family $\mathrm{cRat}(\mathbb{R})$ of subsets of $\mathbb{R}$ satisfying

- $\emptyset$ and the singletons $\{t\}$, $t \in \mathbb{M}$, belong to $\mathrm{cRat}(\mathbb{R})$.
- If $A \subseteq \mathbb{M}$ and $B, C \subseteq \mathbb{R}$ are in $\mathrm{cRat}(\mathbb{R})$, then also the *product* $A\,B$ and the *union* $B \cup C$ belong to $\mathrm{cRat}(\mathbb{R})$.
- If $A \subseteq \mathbb{M}$ is in $\mathrm{cRat}(\mathbb{R})$, then also the *concurrent iteration* A^{c*} and the *concurrent ω-iteration* $A^{c\omega}$ belong to $\mathrm{cRat}(\mathbb{R})$.

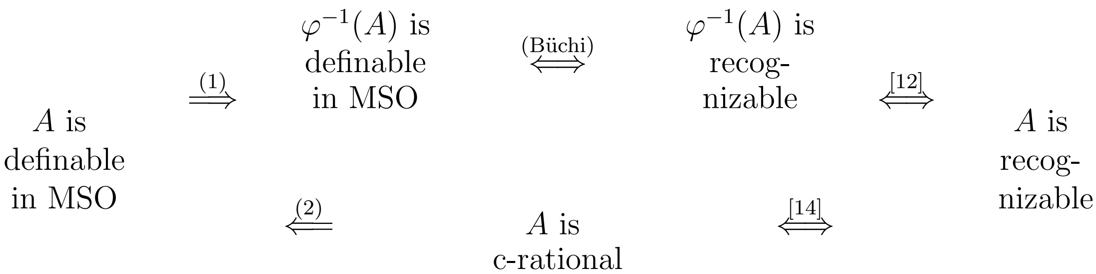
Actually we shall use in the following a slight modification of the above definition, which is easily shown to be equivalent [14]: we replace the closure by the two concurrent iteration operators by the following closure property:

- If $A \subseteq \mathbb{M}$ is *connected* and belongs to $\mathrm{cRat}(\mathbb{R})$, then also the *Kleene-iteration* A^* and the *ω-iteration* A^ω belong to $\mathrm{cRat}(\mathbb{R})$.

In the proof of the theorem below we use the equivalence between recognizability and MSO definability for ω-word languages [4]. Furthermore, the proof is based on the equivalence between $\mathrm{Rec}(\mathbb{R})$ and $\mathrm{cRat}(\mathbb{R})$ [14].

Theorem 2 *Let $A \subseteq \mathbb{R}$ be a real trace language. Then A is recognizable if and only if it is definable in monadic second-order logic.*

Proof: The figure below sketches the situation considered. Recall that φ denotes the canonical mapping $\varphi : \Sigma^\infty \twoheadrightarrow \mathbb{R}$.



(1): Clearly, we can not use the same formula for both models, traces and words, as the underlying interpretation of the predicate $<$ is different. For example the formula

$$\exists x\, \exists y\, \exists z\, \Big(P_a(x) \wedge P_c(y) \wedge P_b(z) \wedge x < z \wedge y < z \wedge (x < y \vee y < x)\Big)$$

over the dependence alphabet $(\Sigma, D) = \quad a — b — c \quad$ is not true for the trace acb, but it is true for both representatives acb and cab of this trace. However, it suffices to express the partial order on traces by the linear order on words.

Given a MSO formula ψ defining $A \subseteq \mathbb{R}$, we replace every subformula $x < y$ in ψ by a subformula $x <_{lin} y$ given below, obtaining a MSO formula ψ' satisfying for all $t \in \mathbb{R}$: $t \models \psi$ if and only if $w \models \psi'$, for all $w \in \Sigma^\infty$ with $t = \varphi(w)$.

Note that in any dependence graph $[V, E, \ell]$, we have for vertices $x, y \in V$: $x < y$, if and only if there is a sequence $x = x_1 < \cdots < x_k < x_{k+1} = y$, $k \geq 1$, $x_i \in V$, such that $(\ell(x_i), \ell(x_{i+1})) \in D$, for every $1 \leq i \leq k$. Moreover, due to the reflexivity of D, we may clearly restrict to the case $k \leq |\Sigma|$. This yields the following definition of $x <_{lin} y$:

$$\bigvee_{1 \leq k \leq |\Sigma|} \quad \bigvee_{\substack{a_1,\ldots,a_{k+1} \in \Sigma \\ \text{with } (a_i, a_{i+1}) \in D \\ \text{for } 1 \leq i \leq k}} \exists x_2 \ldots \exists x_k \Big(P_{a_1}(x) \wedge P_{a_2}(x_2) \wedge \ldots \wedge P_{a_k}(x_k) \wedge P_{a_{k+1}}(y) \wedge x < x_2 < \cdots < x_k < y\Big) \;.$$

(2): This implication is shown by induction over c-rational expressions. Note that we consider both finite and infinite traces as models. Every formula ψ_A given below which defines a real trace language $A \subseteq \mathbb{R}$ can be expressed as the disjunction of subformulae satisfied either only by finite or only by non-finite traces.

- For $A = \emptyset$ let $\psi_\emptyset = \exists x (x < x)$ define A; for $A = \{t\}$, $t \in \mathbb{M}$, let $\psi_{\{t\}}$ be a formula satisfied by the trace $t \in \mathbb{M}$, only.
- $A \cup B$ for c-rational sets A and B: Combine the formulae ψ_A and ψ_B for A and B to $\psi_{A \cup B} = \psi_A \vee \psi_B$.
- $A \cdot B$ for c-rational sets $A \subseteq \mathbb{M}$ and $B \subseteq \mathbb{R}$, defined by ψ_A, respectively ψ_B: For a formula ψ we use in the following relativizations $\psi|_R$ of ψ with respect to a unary predicate R. Recall the inductive definition of $\psi|_R$: $\psi|_R = \psi$ for atomic formulae ψ; $(\neg\psi)|_R = \neg\psi|_R$, $(\psi_1 \vee \psi_2)|_R = \psi_1|_R \vee \psi_2|_R$, respectively $(\exists x\, \psi)|_R = \exists x (R(x) \wedge \psi|_R)$ and $(\exists X\, \psi)|_R = \exists X (X \subseteq R \wedge \psi|_R)$, where $X \subseteq R$ abbreviates $x \in X \to R(x)$. The unary predicate R used below is given as a set property. Let $\psi_{A \cdot B}$ be defined as

$$\bigvee_{0 \leq k \leq |\Sigma|} \exists x_1 \ldots \exists x_k \Big(\psi_A|_{\{x|\bigvee_{1 \leq i \leq k} x \leq x_i\}} \wedge \psi_B|_{\{x|\bigwedge_{1 \leq i \leq k} \neg x \leq x_i\}}\Big) \;.$$

It is easy to see that $\psi_{A\cdot B}$ defines exactly the product $A\cdot B$. The meaning of the variables $x_1,\ldots,x_k$ is to include the maximal vertices of the left factor, which is supposed to belong to A.

- A^*, A^ω for a connected c-rational set $A \subseteq \mathbb{M}$: Let ψ_A denote a sentence defining the connected language $A \subseteq \mathbb{M}$ and assume for sake of simplicity $1 \notin A$.

  The idea for a formula defining the (finite or ω-) iteration of a language A is to colour the vertices of the dependence graph of the considered trace t, such that the colouring corresponds to a factorization in connected factors $t = t_1\, t_2 \ldots$, where every factor $t_i \in \mathbb{M}$ belongs to A. The identification of (A-) factors will be provided by the property of being one-coloured and by the restriction that for any two different factors having the same colour there is no edge of the Hasse diagram connecting them. Two factors t_i, t_j will have the same colour only if $\mathrm{alph}(t_i) = \mathrm{alph}(t_j)$. For every $\Sigma' \subseteq \Sigma$ we take two colours and colour alternatingly the factors t_i with the colours of $\mathrm{alph}(t_i)$.

  We define ψ_{A^*} and ψ_{A^ω} as

  $$\exists X_1 \ldots \exists X_k \ (\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4) \ ,$$

  where $X_1,\ldots,X_k$ are supposed to represent the vertex colouring, with $k = 2\cdot 2^{|\Sigma|}$.

  The formula ψ_1 states that V is the disjoint union of all X_i. For the next two subformulae we define below the abbreviation $mocs(X)$, which expresses that X is a "maximal one-coloured connected subgraph" of the Hasse diagram. Formally, let $mocs(X)$ be

  $$\bigvee_{1\le i\le k} \Big(X \subseteq X_i \quad \wedge \quad \text{“}X \text{ is connected”} \\ \wedge\, \forall y \forall z \, \Big((y \in X \wedge z \in X_i \wedge (z \; E_H \; y \vee y \; E_H \; z)) \rightarrow z \in X \Big)\Big),$$

  where E_H is the edge relation of the Hasse diagram. The (first-order) subformula "X is connected" simply requires that the set of labels occurring in X is a connected subalphabet $\Gamma \subseteq \Sigma$.

  The formula ψ_2 ensures that every *mocs*-component is an element of A and is defined as

  $$\forall X \ (mocs(X) \rightarrow \psi_A|_X) \ ,$$

  where $\psi_A|_X$ denotes the relativization of the sentence defining A with respect to the predicate $x \in X$. Note that since A is a language of finite traces, the *mocs*-components satisfying ψ_2 will be finite.

  The underlying interpretation of the *mocs*-components is that they are factors of the given trace belonging to A. It remains to provide that the *mocs*-components can be ordered, thus corresponding to a factorization. Let us define the relation $X \prec Y$ as

$$\exists x \exists y\ (x \in X \wedge y \in Y \wedge x < y)\ .$$

Due to the reflexivity of the dependence relation D it suffices to forbid cycles of *mocs*-components $Y_1 \prec Y_2 \prec \cdots \prec Y_k \prec Y_{k+1} = Y_1$ with $k \leq |\Sigma|$. Consider otherwise a cycle as above, with $Y_i \neq Y_j$ for $i \neq j$, and vertices $x_i, y_i \in Y_i$ with $x_i < y_{i+1}$, for $1 \leq i \leq k$. For $k > |\Sigma|$ we obtain by considering equal labels in $\{x_1, \ldots x_k\}$ indices $1 \leq i \neq j \leq k$ with $Y_i \prec Y_j$ and $Y_i \prec Y_{j+1}$. It is easy to see that these relations yield a smaller cycle. The formula ψ_3 ensures that the relation $\prec$ restricted on *mocs*-components is acyclic:

$$\neg \bigvee_{2 \leq k \leq |\Sigma|} \exists Y_1 \ldots \exists Y_k \Big(\bigwedge_{1 \leq i \leq k} mocs(Y_i)\ \wedge$$
$$Y_1 \prec Y_2 \wedge \ldots \wedge Y_{k-1} \prec Y_k \wedge Y_k \prec Y_1 \Big)\ .$$

Finally ψ_4 determines whether its trace models are finite or not, depending on the type of iteration (Kleene-star or ω). For the ω-iteration e.g., we have $\psi_4 = \bigvee_{a \in \Sigma} \Big(\forall x (P_a(x) \rightarrow \exists y (x < y \wedge P_a(y))) \Big)$.

□

The proof of Theorem 2 can also be provided using the classical automaton-based approach given by Büchi [24]. Using the characterization of recognizability by nondeterministic asynchronous (cellular) automata with local Büchi acceptance [13], one can view a run of an asynchronous automaton as a labelling of a dependence graph with local states. This allows to show that a trace language $A \subseteq \mathbb{R}$ is accepted by some Büchi asynchronous cellular automaton if and only if A is definable in MSO [11] (respectively [25] for finite traces).

4 First-Order Logic on Real Traces

For (ω-) word languages the restriction of the logical framework to quantifying only over first-order variables turns out to yield a subclass of recognizable languages with various interesting properties. First-order definable (ω-) word languages are closely related to temporal logic of linear time and show to coincide with the star-free languages. Moreover, they can be captured by an algebraic property (aperiodicity) [24, for an overview]. For languages of finite traces we have the equivalence between aperiodicity and star-freeness [15], respectively between star-freeness and first-order definability, as shown by Thomas and Zielonka [personal communication].

The first-order logical framework we consider (denoted by FO) restricts for the present signature the quantification of formulae to variables $x, y, \ldots$, only. In the following we first exhibit a direct transformation between formulae interpreted on finite trace models and formulae interpreted on finite word models (the result also holds for higher order logic). For the general case of real trace languages we obtain the equivalence stated in Proposition 3 below by the characterizations of Sections 4.1, 4.2. A direct construction as presented below is of independent interest.

In the proof of the following proposition we use the same approach for both formulae transformations. In particular, we replace every subformula $x < y$ of a sentence ψ, which is interpreted on words, by a formula suitable for the partial order interpretation. However, it is not possible to unify different interpretations of a formula on the representatives of a trace to one interpretation for the trace model itself. Consider e.g. the formula $\forall x\, \forall y\; (x \leq y \;\vee\; y \leq x)$, which is false on any dependence graph containing two incomparable vertices, but is always true on word models.

The idea is to fix a representative of each trace and to express the total order of the fixed representative by the partial order of the dependence graph. As representative of a trace we choose the lexicographic normal form. (The use of the lexicographic normal form here was observed independently by W. Zielonka.) Given a linear ordering $<_\Sigma$ of Σ, the *lexicographic normal form* of a trace $t \in \mathbb{M}$ (in symbols $lnf(t)$) is lexicographically the first representative $w \in \varphi^{-1}(t)$. Equivalently, a word $w \in \Sigma^*$ is the lexicographic normal form of its trace $\varphi(w)$ if and only if for each factor aub of w with $a, b \in \Sigma$, $u \in \Sigma^*$, and $(au, b) \in I$, we have $a <_\Sigma b$ [2].

Proposition 3 *A trace language $A \subseteq \mathbb{M}$ is definable in first-order logic if and only if $\varphi^{-1}(A) \subseteq \Sigma^*$ is definable in first-order logic.*

Proof: Given a first-order sentence defining $A \subseteq \mathbb{M}$ we obtain directly from the proof of Theorem 2 a first-order sentence defining $\varphi^{-1}(A)$.

Suppose now we are given a first-order sentence ψ defining a closed word language $A = \varphi^{-1}\varphi(A) \subseteq \Sigma^*$.

As outlined above, we replace every subformula $x < y$ occurring in the sentence ψ defining $A \subseteq \Sigma^*$ by the subformula $lex(x, y)$ given below. The new sentence ψ' will satisfy $t \models \psi'$ if and only if $lnf(t) \models \psi$. Due to A being a closed word language, this will yield the result.

Let x, y be vertices in the dependence graph of t and let $lex(x, y)$ denote the predicate expressing that x precedes y in the lexicographic normal form of t, $lnf(t)$. Assume that $lex(x, y)$ holds and let $x_0 \cdots x_n$ correspond to the factor in $lnf(t)$ satisfying $x_0 = x$ and $x_n = y$. Let i be minimal, $0 \leq i \leq n$, such that $x_i \leq y$ holds in the partial order of the dependence graph. Then we have $(x_0 \cdots x_{i-1}, x_i) \in I$ and hence $\ell(x_0) \leq_\Sigma \ell(x_i)$. It is easy to see now that $lex(x, y)$

is equivalent to $\exists z\,(\ell(x) \leq_\Sigma \ell(z) \wedge z \leq y \wedge lex(x,z))$. This observation leads to defining $lex(x,y)$ as $\bigvee_{a,b\in\Sigma}(P_a(x) \wedge P_b(y) \wedge lex_{a,b}(x,y))$, where $lex_{a,b}(x,y)$ is defined recursively as

$$lex_{a,b}(x,y) = \begin{cases} x < y & \text{for } a = b\ , \\ \neg\, lex_{b,a}(y,x) & \text{for } a <_\Sigma b\ , \\ \exists z \left(\bigvee_{c \geq_\Sigma a} (P_c(z) \wedge z \leq y \wedge lex_{a,c}(x,z)) \right) & \text{for } a >_\Sigma b\ . \end{cases}$$

Note that the recursion depth is at most $2 \cdot |\Sigma|$, yielding a first-order formula of exponential size in $|\Sigma|$. The quantifier depth is bounded by $|\Sigma|$. □

Remark 4 The result stated in Proposition 3 clearly also holds if we replace first-order by second-order logic. This, together with Theorem 2, provides a new proof for Métivier's [18, Theorem 2.3] and Ochmański's [19, Lemma 8.2] theorem on the recognizability of the Kleene-iteration of connected recognizable finitary trace languages.

Remark 5 Since the lexicographic normal form is in general undefined for real traces, the above proof can be extended directly to real traces only for the special case where the considered language contains only traces t where the set of letters occurring infinitely often in t is a connected subalphabet.

4.1 Equivalence of Star-Free Expressions and First-Order Logic

In this section we show the equivalence of star-freeness and definability in first-order logic. We generalize the approach of Perrin and Pin [21] from words to real traces.

The family of *star-free* finitary trace languages $\mathrm{SF}(\mathbb{M})$ is the closure of the sets $\{t\}$, $t \in \mathbb{M}$, by Boolean operations and concatenation [15].

Definition 6 *The family* $\mathrm{SF}(\mathbb{R})$ *of* star-free *real trace languages is the smallest family* $\mathcal{F}$ *of subsets of* $\mathbb{R}$ *with*

1. $\mathrm{SF}(\mathbb{M}) \subseteq \mathcal{F}$,
2. $AB \in \mathcal{F}$ *for any* $A, B \in \mathcal{F}$ *with* $A \subseteq \mathbb{M}$, $B \subseteq \mathbb{R}$, *and*
3. $\mathcal{F}$ *is closed under Boolean operations (where the complementation is meant with respect to* $\mathbb{R}$*).*

Note that since $\mathbb{M} \in \mathrm{SF}(\mathbb{R})$ holds, the complementation of a finitary language $A \in \mathrm{SF}(\mathbb{M})$ with respect to $\mathbb{M}$ can be obtained by complementing with respect to $\mathbb{R}$ and intersecting with $\mathbb{M}$.

We will consider formulae with *free variables* and interpret them on extended trace models. For a finite set W of variables we add to a structure $(V, <, (P_a)_{a\in\Sigma})$

a mapping of variables $\sigma : W \to V$, associating every variable with one vertex in the dependence graph of the trace. Thus we obtain a new structure $(V, <, (P_a)_{a\in\Sigma}, \sigma)$ of a trace with a set W of variables, denoted W-trace. Let a set of W-traces be denoted as a W-trace language. If W is empty, then the new structure $(V, <, (P_a)_{a\in\Sigma}, \emptyset)$ can be identified with the former one, $(V, <, (P_a)_{a\in\Sigma})$. We denote by $\mathbb{M}_W$ (resp. $\mathbb{R}_W$) the set of finite W-traces (resp. real W-traces).

The concatenation of a W-trace t and a W'-trace t' is defined only if $W \cap W' = \emptyset$; in this case, in addition to the usual concatenation of dependence graphs, the mapping of variables of tt' is the disjoint union of the mapping of variables of t and t'. The complement of a real W-trace language $A \subseteq \mathbb{R}_W$ is meant with respect to $\mathbb{R}_W$. The definition of star-free languages can now be immediately extended to star-free real trace languages over extended structures. Let $\mathrm{SF}_e(\mathbb{M})$ denote the family of W-trace languages (for finite sets W) obtained from the sets $\{t\}$ ($t \in \mathbb{M}_W$ for finite W) by concatenation and Boolean operations, where the complement of a W-trace language is meant with respect to $\mathbb{M}_W$. Further, let $\mathrm{SF}_e(\mathbb{R})$ denote the smallest family of subsets of $\mathbb{R}_W$ (for finite sets W) containing $\mathrm{SF}_e(\mathbb{M})$ and closed under concatenation and Boolean operations.

For the rest of the section, sets W of (first-order) variables are supposed to be finite. The following lemma states some useful properties of star-free trace languages over the extended structures defined above. The left and right quotients $t^{-1}A$ and At^{-1} for $t \in \mathbb{R}_U$, $A \subseteq \mathbb{R}_W$ are defined as usual, by $t^{-1}A = \{ v \mid tv \in A \}$ and $At^{-1} = \{ v \mid vt \in A \}$. Note that the partial monoid of real traces $\mathbb{R}$ is left-, but not right-cancellative: If uv, uv', vu, and $v'u$ are defined, we have $uv = uv' \Rightarrow v = v'$, but $vu = v'u$ does not imply $v = v'$. In particular it does not hold that $\{t\}t^{-1} = \{1\}$, for example $\{a^\omega\}(a^\omega)^{-1} = a^* \neq \{1\}$.

Lemma 7 *1. For any $\Sigma' \subseteq \Sigma$ and any set W, the languages $\{t \in \mathbb{M}_W \mid \mathrm{alph}(t) \subseteq \Sigma'\}$ respectively $\{t \in \mathbb{R}_W \mid \mathrm{alph}(t) \subseteq \Sigma'\}$ are star-free.*

2. Taking left and right quotients commutes with $\cup$ and $\cap$. Moreover, let $t \in \mathbb{R}_U$ and $A \subseteq \mathbb{R}_W$ with $U \subseteq W$. Then we have $\overline{t^{-1}A} = t^{-1}\overline{A} \cup \{u \in \mathbb{R}_{W\setminus U} \mid tu \notin \mathbb{R}_W\}$, respectively $\overline{At^{-1}} = \overline{A}t^{-1} \cup \{u \in \mathbb{R}_{W\setminus U} \mid ut \notin \mathbb{R}_W\}$.

3. Let $A \in \mathrm{SF}_e(\mathbb{R})$. Then the set of left (right) quotients $\{ t^{-1}A \mid t \in \mathbb{R}_U, U \}$ ($\{ At^{-1} \mid t \in \mathbb{R}_U, U \}$) is a finite subset of $\mathrm{SF}_e(\mathbb{R})$.

Proof: 1: Obvious.
2: Note that for $u \in \mathbb{R}_{W\setminus U}$ we have $u \notin t^{-1}A$ if and only if either $tu \in \overline{A}$ or tu is not a real trace.
3: Assume $A \in \mathrm{SF}_e(\mathbb{R}) \cap \mathbb{R}_W$. The proof is given by induction on the star-free expression for A. We denote the number of different languages $t^{-1}A$ by $n(A)$.
If $A = \{u\}$ for some $u \in \mathbb{M}_W$, then $n(A) \leq 2^{|u|}$.
If $A = B \cup C$, then $t^{-1}A = t^{-1}B \cup t^{-1}C$, hence $n(A) \leq n(B) \cdot n(C)$ ($\cap$ analogously).
If $A = \overline{B}$, then with $t^{-1}\overline{B} = \overline{t^{-1}B} \setminus \{v \in \mathbb{R}_{W\setminus U} \mid \mathrm{alph}(v) \cap D(\mathrm{alphinf}(t)) \neq \emptyset\}$ we obtain $n(A) \leq n(B) \cdot 2^{|\Sigma|}$.

If $A = BC$, then $t^{-1}(BC) = \bigcup_{rs=t}(r^{-1}B \cap F_s)(s^{-1}C)$, where $F_s = \{p \in \mathbb{R}_{W'} \mid (p,s) \in I\} \in \mathrm{SF}_e(\mathbb{R})$ (for some $W' \subseteq W$), thus

$$\{\, t^{-1}(BC) \mid t \in \mathbb{R}_U \,\} = \{\, \bigcup_{rs=t} (r^{-1}B \cap F_s)(s^{-1}C) \mid t \in \mathbb{R}_U \,\} \enspace ,$$

with $n(r^{-1}B \cap F_s) \leq n(B) \cdot 2^{|\Sigma|}$ and $n(A) \leq 2^{n(B)\, 2^{|\Sigma|}\, n(C)}$.
Right quotients At^{-1} for $A \in \mathrm{SF}_e(\mathbb{R})$ are handled symmetrically. □

Theorem 8 *A trace language $A \subseteq \mathbb{R}$ is definable in first-order logic if and only if it is star-free.*

Proof: "⇐": Set operations are replaced by the corresponding logical operations. For the concatenation we use the first-order formula from Section 3.

"⇒": We give a proof by induction on formulae.

Predicates: The set of real W-traces satisfying $x < y$ for some variables $x, y \in W$, is a star-free trace language of the following form (we omit the subscripts for $\mathbb{M}, \mathbb{R}$ and $a \in \Sigma_X$ denotes that the set of variables X is assigned to vertex a):

$$\bigcup_{\text{finite}} \quad \bigcup_{\substack{a_1 \in \Sigma_X, a_l \in \Sigma_Y,\ x \in X,\ y \in Y \\ \text{with } (a_i, a_{i+1}) \in D \text{ for } 1 \leq i \leq l-1}} \mathbb{M} a_1 \mathbb{M} a_2 \cdots a_{l-1} \mathbb{M} a_l \mathbb{R} \enspace .$$

Note that the sets of variables assigned to the factors in the expression above form a partition of W and $l \leq |\Sigma|$.

A real W-trace language satisfying $P_a(x)$ is of the form $\bigcup_{\text{finite}} \mathbb{M} a \mathbb{R}$ with $a \in \Sigma_X$ and $x \in X$. For $x = y$ we have the representation $\bigcup_{\text{finite}} \bigcup_{\substack{a \in \Sigma_X \\ x,y \in X}} \mathbb{M} a \mathbb{R}$.

Formulae: For $\wedge$, $\vee$, $\neg$ we use the corresponding set operations. Finally, for quantified formulae, if $A \in \mathrm{SF}_e(\mathbb{R}) \cap \mathbb{R}_W$ is the language defined by a formula ψ with free variables W and $x \in W$, then we first express A using left and right quotients [21] (we omit again the subscript for $\mathbb{R}$)

$$A \quad = \bigcup_{\substack{a \in \Sigma_X,\ ua \in \mathbf{R}, \\ x \in X \subseteq W}} B(u)\, a\, C(u)$$

with

$$C(u) = (ua)^{-1}A, \quad B(u) = \bigcap_{v \in C(u)} A(av)^{-1} \enspace .$$

By Lemma 7 the union above is finite. Finally, the real $W \setminus \{x\}$-trace language A' defined by the formula $\exists x\ \psi$ is

$$A' = \bigcup_{a \in \Sigma_{X \setminus \{x\}},\ ua \in \mathbf{R}} B(u)\, a\, C(u)$$

□

4.2 Aperiodic and Star-Free Real Trace Languages

A monoid is called *aperiodic* if it satisfies the equation $x^n = x^{n+1}$ for some $n > 0$. Let $A \subseteq \mathbb{R}$ be a real trace language. A is called *aperiodic* if there exists a homomorphism $\eta : \mathbb{M} \to S$ to an aperiodic, finite monoid S recognizing A (Equivalently, the syntactic monoid of A is finite and aperiodic, and the syntactic morphism recognizes A.).

We denote the family of aperiodic real (finitary, respectively) trace languages by AP($\mathbb{R}$) (AP($\mathbb{M}$), respectively). We use analogous notations for word languages. We recall Schützenberger's result stating the equivalence between aperiodicity and star-freeness for finitary word languages [23]. The result has been extended to ω-word languages by Perrin [20] and for languages of finite traces by Guaiana et. al. [15].

Let us begin with some notations concerning recognizability by homomorphisms and consider $\eta : \mathbb{M} \to S$ to a finite monoid S. For $s \in S$ we denote

$$M_s = \eta^{-1}(s)$$
$$P_s = M_s \setminus M_s\mathbb{M}_+, \quad \text{with } \mathbb{M}_+ = \mathbb{M} \setminus \{1\} \ .$$

Thus, $\mathbb{M}_s$ is the set of all finite traces which are mapped to s by η and P_s is the subset of $\mathbb{M}_s$, consisting of traces having no proper prefix in $\mathbb{M}_s$. Finally, if we consider a homomorphism $\eta : \Sigma^* \to S$, then we use the notation $X_s = \eta^{-1}(s)$, for $s \in S$.

Moreover, we may assume that $\text{alph}(t) = \text{alph}(t')$ for all $t, t' \in \mathbb{M}$ with $\eta(t) = \eta(t')$, since we may replace S with a submonoid of $S \times \mathcal{P}(\Sigma)$, with the multiplication defined by $(s, \Gamma)(s', \Gamma') = (ss', \Gamma \cup \Gamma')$ and $(1, \emptyset)$ as identity. Moreover we replace $\eta(a)$ with $(\eta(a), \{a\})$ for $a \in \Sigma$. Hence, $\text{alph}(s)$ for $s \in S$ can be defined as $\text{alph}(t)$ for some $t \in \eta^{-1}(s)$. Note that any aperiodic monoid S remains aperiodic if we replace it with (a submonoid of) $S \times \mathcal{P}(\Sigma)$.

For $\Sigma' \subseteq \Sigma$, let $\mathbb{R}_{\Sigma'} = \{ t \in \mathbb{R} \mid D(\text{alphinf}(t)) = D(\Sigma') \}$. Note that in the word case (i. e. $D = \Sigma \times \Sigma$) we have $\mathbb{R}_{\Sigma'} = \Sigma^\omega$, for every $\emptyset \neq \Sigma' \subseteq \Sigma$ and $\mathbb{R}_\emptyset = \Sigma^*$. In particular, we denote by $\mathbb{R}_s$ for $s \in S$ the set $\mathbb{R}_{\Sigma'}$ with $\Sigma' = \text{alph}(s)$.

Let $A \subseteq \mathbb{M}$. We define $\overrightarrow{A} = \{ t \in \mathbb{R} \mid t = \sqcup B \text{ with } B \text{ directed and } B \subseteq A \}$. A non-empty set $B \subseteq \mathbb{M}$ is called directed if for every $t, t' \in B$, there exists a $z \in B$ such that t and t' are both prefixes of z.

The following lemma generalizes a lemma used in Schützenberger's proof of McNaughton's theorem for ω-word languages [22]. For real traces, the proof of the lemma becomes more involved and the reader is referred directly to [9].

Lemma 9 *[9] Let S be a finite monoid, $\eta : \mathbb{M} \to S$ a homomorphism and $e \in S$ such that $e^2 = e$. Then we have $\mathbb{M}_e^\omega = \overrightarrow{\mathbb{M}_e P_e} \cap \mathbb{R}_e$.*

Before stating the result of this section, let us define the I-shuffle $K_1 \sqcup\!\sqcup_I K_2$ of two (ω-) word languages $K_i \subseteq \Sigma^\infty$ by $K_1 \sqcup\!\sqcup_I K_2 = \{ u_0 v_0 u_1 v_1 \ldots \mid u_n, v_n \in \Sigma^*,\ u_0 u_1 \ldots \in K_1,\ v_0 v_1 \ldots \in K_2 \text{ and } (v_n, u_m) \in I, \text{ for } n < m \}$.

Theorem 10 *The family of star-free real trace languages coincides with the family of aperiodic real trace languages.*

Proof: "$\Leftarrow$": Let us first show that every aperiodic language $A \subseteq \mathbb{R}$ is star-free. To this purpose, consider $\eta : \mathbb{M} \to S$ a homomorphism to a finite, aperiodic monoid S recognizing A. Then we have $A = \bigcup_{(s,e)\in P} \mathbb{M}_s \mathbb{M}_e^\omega$, with $P = \{ (s,e) \in S^2 \mid se = s,\ e^2 = e,\ \mathbb{M}_s\mathbb{M}_e^\omega \cap A \neq \emptyset \}$. Since $\mathbb{M}_s \in \mathrm{SF}(\mathbb{M})$ [15], it suffices to show that $\mathbb{M}_e^\omega \in \mathrm{SF}(\mathbb{R})$. By Lemma 9, $\mathbb{M}_e^\omega = \overrightarrow{\mathbb{M}_e P_e} \cap \mathbb{R}_e$. Furthermore, $\mathbb{R}_e \in \mathrm{SF}(\mathbb{R})$. More precisely, $\mathbb{R}_e$ (with $e \neq 1$) is a finite union of sets $\mathbb{M}\mathbb{M}_{\Sigma'}^\omega$ with $\Sigma' \subseteq \Sigma$, $\mathbb{M}_{\Sigma'} = \{t \in \mathbb{M} \mid \mathrm{alph}(t) = \Sigma'\}$ and $\mathbb{M}_{\Sigma'}^\omega = \left(\bigcap_{b\notin\Sigma'} \overline{\mathbb{M}b\mathbb{R}} \cap \bigcap_{a\in\Sigma'} \overline{\mathbb{M}\overline{\mathbb{M}a\mathbb{R}}}\right) \setminus \mathbb{M}$, hence the result.

It remains to show $\overrightarrow{\mathbb{M}_e P_e} \in \mathrm{SF}(\mathbb{R})$. More generally, if $B \subseteq \mathbb{M}$ is recognized by a homomorphism $\eta : \mathbb{M} \to S$ to a finite monoid S, then we may write the complement of $\overrightarrow{B}$ as follows (analogously to [20]):

$$\overline{\overrightarrow{B}} = \bigcup_{p\in S} (\mathbb{M}_p \overline{\bigcup_{q \text{ with } pq\in\eta(B)} \mathbb{M}_q \mathbb{R}}) \ .$$

The above expression simply states that $t \notin \overrightarrow{B}$ if and only if there exists a finite prefix $u \leq t$ such that for every $v \in \mathbb{M}$ with $uv \leq t$, $uv \notin B$. Moreover, with S aperiodic we obtained $\overrightarrow{B} \in \mathrm{SF}(\mathbb{R})$ by [15].

"$\Rightarrow$": For this inclusion, it suffices to show that $A \in \mathrm{SF}(\mathbb{R})$ implies $\varphi^{-1}(A) \in \mathrm{SF}(\Sigma^\infty)$, since $\mathrm{SF}(\Sigma^\infty) = \mathrm{AP}(\Sigma^\infty)$ [20] and $\mathrm{Synt}(A) = \mathrm{Synt}(\varphi^{-1}(A))$. We proceed by induction on the star-free expression denoting $A \in \mathrm{SF}(\mathbb{R})$. For $A \in \mathrm{SF}(\mathbb{M})$ we have $A \in \mathrm{AP}(\mathbb{M})$ by [15], hence $\varphi^{-1}(A) \in \mathrm{AP}(\Sigma^*) = \mathrm{SF}(\Sigma^*)$ [23]. Furthermore, let $A = A_1 \cup A_2$ ($A = A_1 \cap A_2$, $A = \overline{A_1}$ respectively) with $\varphi^{-1}(A_1), \varphi^{-1}(A_2) \in \mathrm{SF}(\Sigma^\infty)$. Then $\varphi^{-1}(A) \in \mathrm{SF}(\Sigma^\infty)$ holds, since φ^{-1} commutes with the Boolean operations.

Finally, let $A = A_1A_2$ with $A_1' = \varphi^{-1}(A_1) \in \mathrm{SF}(\Sigma^*)$, $A_2' = \varphi^{-1}(A_2) \in \mathrm{SF}(\Sigma^\infty)$. Then, $\varphi^{-1}(A) = A_1' \sqcup\!\sqcup_I A_2'$, with $\sqcup\!\sqcup_I$ denoting the I-shuffle operation. In particular, we have

$$\begin{aligned} A_1' \sqcup\!\sqcup_I A_2' \ = \ \{ & u_0v_0 \ldots u_nv_nw \mid u_k, v_k \in \Sigma^*,\ w \in \Sigma^\infty,\ u_0u_1 \ldots u_n \in A_1', \\ & v_0v_1 \ldots v_nw \in A_2' \text{ and } (v_i, u_k) \in I, \text{ for } i < k \leq n \} \ . \end{aligned}$$

Since $A_1' \in \mathrm{SF}(\Sigma^*) = \mathrm{AP}(\Sigma^*)$ and $A_2' \in \mathrm{SF}(\Sigma^\infty) = \mathrm{AP}(\Sigma^\infty)$, let $\eta : \Sigma^* \to S$ denote a homomorphism recognizing both A_1' and A_2', with S an aperiodic, finite monoid (for A_2' this means that η recognizes $A_2' \cap \Sigma^\omega$ and $A_2' \cap \Sigma^*$). Furthermore, we consider the set $P = \{ (s,e) \in S^2 \mid se = s,\ e^2 = e,\ X_sX_e^\omega \cap A_2' \neq \emptyset \}$. Noting that $X_1 = \{1\} = X_1^\omega$, with 1 denoting the identity in S, we have $A_2' = \bigcup_{(s,e)\in P} X_sX_e^\omega$. Moreover, $A_1' = \eta^{-1}\eta(A_1')$. It is not hard to see that we may express $A_1' \sqcup\!\sqcup_I A_2'$ by

$$A_1' \sqcup\!\sqcup_I A_2' = \bigcup_{r\in\eta(A_1')} \bigcup_{(s,e)\in P} (X_r \sqcup\!\sqcup_I X_s) X_e^\omega \ .$$

Since both X_r and X_s are aperiodic, closed finitary word languages (due to A'_i being closed), we have $X_r \sqcup\!\sqcup_I X_s \in \mathrm{SF}(\Sigma^*)$ [15]. Moreover, since e is an idempotent element of S, with S aperiodic, we also have $X_e^\omega \in \mathrm{SF}(\Sigma^\infty)$ [20]. Hence, $A'_1 \sqcup\!\sqcup_I A'_2 = \varphi^{-1}(A) \in \mathrm{SF}(\Sigma^\infty)$. □

Let us summarize: Since for every $A \in \mathrm{Rec}(\mathbb{R})$, $\mathrm{Synt}(A) = \mathrm{Synt}(\varphi^{-1}(A))$ holds, we obtained by the results of Section 4 the following equivalent characterizations:

i) A is first-order definable.

ii) $\varphi^{-1}(A)$ is first-order definable.

iii) A is star-free.

iv) $\mathrm{Synt}(A)$ is aperiodic and the syntactic homomorphism $\eta : \mathbb{M} \to \mathrm{Synt}(A)$ recognizes A.

5 Conclusion

In this paper we have generalized the most important results concerning logic, recognizability, star-freeness and aperiodicity from word languages and finitary trace languages to the case of infinitary trace languages.

For star-free languages an extension to the quantifier alternation hierarchy of first-order formulae and dot-depth hierarchies of languages has been considered [11, contains also a temporal logic which is expressively equivalent to first-order logic for finite traces].

Acknowledgements

We want to thank V. Diekert, P. Gastin, H. J. Hoogeboom, D. Kuske, and W. Thomas for many interesting comments. We also thank the anonymous referees from ICALP and especially from Theoretical Computer Science for the valuable comments which helped improving the presentation of this paper.

References

[1] IJ. J. Aalbersberg and G. Rozenberg. Theory of traces. *Theoretical Computer Science*, 60:1–82, 1988.

[2] A. V. Anisimov and D. E. Knuth. Inhomogeneous sorting. *International Journal of Computer and Information Sciences*, 8:255–260, 1979.

[3] A. Arnold. A syntactic congruence for rational ω-languages. *Theoretical Computer Science*, 39:333–335, 1985.

[4] J. R. Büchi. On a decision method in restricted second order arithmetic. In E. Nagel et al., editors, *Proc. Internat. Congress on Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford Univ. Press, Stanford, CA, 1960.

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

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

[7] V. Diekert. On the concatenation of infinite traces. *Theoretical Computer Science*, 113:35–54, 1993. Special issue STACS'91.

[8] V. Diekert and W. Ebinger, editors. *Infinite Traces. Proceedings of a workshop of the ESPRIT Basic Research Action No 3166: Algebraic and Syntactic Methods in Computer Science (ASMICS), Tübingen, Germany, 1992*, Bericht 4/92. Universität Stuttgart, Fakultät Informatik, 1992.

[9] V. Diekert and A. Muscholl. Deterministic asynchronous automata for infinite traces. *Acta Informatica*, 31:379–397, 1994. A preliminary version was presented at STACS'93, Lecture Notes in Computer Science 665 (1993).

[10] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, Singapore, 1995.

[11] W. Ebinger. *Charakterisierung von Sprachklassen unendlicher Spuren durch Logiken*. Dissertation, Institut für Informatik, Universität Stuttgart, 1994.

[12] P. Gastin. Recognizable and rational trace languages of finite and infinite traces. In C. Choffrut 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.

[13] P. Gastin and A. Petit. Asynchronous automata for infinite traces. In W. Kuich, editor, *Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP'92), Vienna (Austria) 1992*, number 623 in Lecture Notes in Computer Science, pages 583–594, Berlin-Heidelberg-New York, 1992. Springer.

[14] P. Gastin, A. Petit, and W. Zielonka. An extension of Kleene's and Ochmański's theorems to infinite traces. *Theoretical Computer Science*, 125:167–204, 1994. A preliminary version was presented at ICALP'91, Lecture Notes in Computer Science 510 (1991).

[15] G. Guaiana, A. Restivo, and S. Salemi. Star-free trace languages. *Theoretical Computer Science*, 97:301–311, 1992. A preliminary version was presented at STACS'91, Lecture Notes in Computer Science 480 (1991).

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

[17] 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.

[18] Y. Métivier. Une condition suffisante de reconnaissabilité dans un monoïde partiellement commutatif. *R.A.I.R.O. — Informatique Théorique et Applications*, 20:121–127, 1986.

[19] E. Ochmański. Regular behaviour of concurrent systems. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 27:56–67, Oct 1985.

[20] D. Perrin. Recent results on automata and infinite words. In M. P. Chytil and V. Koubek, editors, *Proceedings of the 11th Symposium on Mathematical Foundations of Computer Science (MFCS'84), Praha (CSFR) 1984*, number 176 in Lecture Notes in Computer Science, pages 134–148. Springer, Berlin-Heidelberg-New York, 1984.

[21] D. Perrin and J.-E. Pin. First-order logic and star-free sets. *Journal of Computer and System Sciences*, 32:393–406, 1986.

[22] D. Perrin and J. E. Pin. Mots Infinis. Tech. Rep. LITP 93.40, Université Paris 7 (France), 1993. Book to appear.

[23] M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.

[24] W. Thomas. Automata on infinite objects. In J. v. Leeuwen, editor, *Handbook of Theoretical Computer Science*, chapter 4, pages 133–191. Elsevier Science Publishers B. V., 1990.

[25] W. Thomas. On logical definability of trace languages. In V. Diekert, editor, *Proceedings of a workshop of the ESPRIT Basic Research Action No 3166: Algebraic and Syntactic Methods in Computer Science (ASMICS), Kochel am See, Bavaria, FRG (1989)*, Report TUM-I9002, Technical University of Munich, pages 172–182, 1990.

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

[27] W. Zielonka. Safe executions of recognizable trace languages by asynchronous automata. In A. R. Mayer et al., editors, *Proceedings of the 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.