# 10.10 Appendix: Logical Definability of Trace Languages

Werner Ebinger
Universität Stuttgart, Institut für Informatik
Breitwiesenstr. 20–22, 70565 Stuttgart, Germany
ebinger@informatik.uni-stuttgart.de

In this appendix we show the equivalence between recognizability and monadic second order logic definability, and the equivalence of star-freeness and first order logic for trace languages. This connects logic on traces to the theory of recognizable trace languages (Chapters 6, 7, 8, and 11) and extends classical results of Büchi, McNaughton, Ladner and Thomas on word languages to traces. Monadic second order logic on traces has been first considered by Thomas. This section is based on a joint work with Muscholl [91] and on the dissertation of the author [90].

## 10.10.1 First Order Logic and Monadic Second Order Logic on Traces

The underlying structures for the logical setting considered in the following are single (finite or infinite) traces. A finite trace $t \in \mathbb{M}(\Sigma, D)$, respectively an infinite (real) trace $t \in \mathbb{R}(\Sigma, D)$ (Chapter 11), can be represented as a labelled partial order $(V, <, \ell)$, corresponding to the dependence graph of $t$ (see Chapter 1 and 2). A logical formula is interpreted over a *structure* with signature $(V, <, (P_a)_{a\in\Sigma})$, where $P_a$ is the set of vertices labelled by $a$. We allow the empty structure $V = \emptyset$ in order to include the empty trace.

First order formulae and monadic second order formulae are based on first order variables $x, y, z, \ldots$ ranging over $V$ and monadic second order variables $X, Y, Z, \ldots$ ranging over the power set $\mathcal{P}(V)$. First order predicates of the form $x < y$, $P_a(x)$ ($a \in \Sigma$) and monadic second order predicates of the form $X(y)$ are the atomic formulae. Formulae are built inductively using the logical operators $\wedge$ (and), $\vee$ (or), and $\neg$ (not), as well as quantification $\exists$, $\forall$ of first order and second order variables. Formulae $\psi$ defined in this way are called *monadic second order formulae* (denoted by $\psi \in \mathcal{L}_2(\Sigma, D)$). A formula $\psi$ is called *first order formula* (denoted by $\psi \in \mathcal{L}_1(\Sigma, D)$) if it contains no second order variable.

A real trace $t$ is a model for a sentence (i. e. a formula without free variables) $\psi$, if $\psi$ is satisfied by $t$ under the canonical interpretation (denoted by $t \models \psi$). This means that variables are mapped to (sets of) vertices of the dependence graph of $t$. The predicate $P_a(x)$ is "$x$ is labelled with $a \in \Sigma$", the relation $<$ is interpreted as the partial order of the dependence graph and $X(y)$ is "$y$ belongs to $X$".

We also write $y \in X$ instead of $X(y)$ and freely use abbreviations like $\rightarrow$, $\leftrightarrow$, $X \subseteq Y$ for $\forall x\ (x \in X \rightarrow x \in Y)$ and $x = y$ for $\bigvee_{a \in \Sigma} \big(P_a(x) \wedge P_a(y) \wedge \neg(x < y) \wedge \neg(y < x)\big)$.

For the special case of complete dependency $D = \Sigma \times \Sigma$, we obtain the monadic second order logic $\mathcal{L}_2(\Sigma)$ and the first order logic $\mathcal{L}_1(\Sigma)$ on (finite and infinite) words.

**Example 10.10.1** We can express the property "$t$ contains the trace factor $ab$" by

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

The power of monadic second order logic on traces does not depend on whether the $<$-relation or the edge relation $E$ of the Hasse diagram of $(V, <, \ell)$ are chosen to be incorporated in the logic, since they can be expressed one into another. The edge relation $E$ constitutes some kind of successor relation and is expressible by the partial order relation $<$ (even in first order logic):

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

and vice versa (in monadic second order logic):

$$\begin{aligned} x < y \quad \text{iff} \quad \exists x' \Big( & x\ E\ x' \ \wedge \\ & \forall X(x' \in X \wedge \forall z \forall z'(z \in X \wedge z\ E\ z' \rightarrow z' \in X) \\ & \qquad \rightarrow y \in X)\Big) \ . \end{aligned}$$

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

### 10.10.2 Characterizations for Finite Traces

The truth value of a formula depends on whether it is interpreted on words or on traces. For example the formula $\forall x\ \forall y\ (x \leq y \ \vee \ y \leq x)$ is not satisfied by any dependence graph which contains an anti-chain (i. e. a set of pairwise incomparable vertices), but it is always true in the word case, i. e. when $<$ is interpreted as a linear order. Our first result connects the logical definability of finitary word languages to the logical definability of finitary trace languages. By $\varphi : \Sigma^* \rightarrow \mathbb{M}(\Sigma, D)$ we denote the canonical homomorphism associating to a word the corresponding trace.

**Theorem 10.10.2** *Let $A \subseteq \mathbb{M}(\Sigma, D)$ be a language of finite traces. Then the following holds:*

1. *The language $A$ is definable in first order logic $\mathcal{L}_1(\Sigma, D)$ if and only if $\varphi^{-1}(A)$ is definable in first order logic $\mathcal{L}_1(\Sigma)$.*
2. *The language $A$ is definable in second order logic $\mathcal{L}_2(\Sigma, D)$ if and only if $\varphi^{-1}(A)$ is definable in second order logic $\mathcal{L}_2(\Sigma)$.*

**Proof:** In both cases, it suffices to express the partial order on traces by the linear order on words and vice versa (in first-order logic).

"$\Rightarrow$": In the following we denote the linear order on words by $<_{lin}$. For the formula below note that if there is a path in the dependence graph from a vertex $x$ to a vertex $y$, then there is also a path with distinct labellings (up to the fact that $x$ and $y$ may have the same labelling). $x < y$ if and only if

$$\bigvee_{1\leq l\leq|\Sigma|} \bigvee_{\substack{a_1,\ldots,a_l,a_{l+1}\in\Sigma \\ \text{with } (a_i,a_{i+1})\in D \\ \text{for } 1\leq i\leq l}} \exists x_2\ldots\exists x_l \Big(P_{a_1}(x)\wedge P_{a_2}(x_2)\wedge\ldots\wedge P_{a_l}(x_l)\wedge P_{a_{l+1}}(y)\wedge x <_{lin} x_2 <_{lin} \ldots <_{lin} x_l <_{lin} y\Big) \; .$$

"$\Leftarrow$": In order to interpret a formula for words on traces, we view the trace $t$ as one of its representing words, since the formula from $\mathcal{L}_1(\Sigma)$ or $\mathcal{L}_2(\Sigma)$ is satisfied either by all words $w$ with $\varphi(w) = t$ or by none. As representing word we consider the lexicographic normal form of $t$. Given a linear ordering $<_\Sigma$ of $\Sigma$, the lexicographically first representing word $w \in \varphi^{-1}(t)$ is called the *lexicographic normal form* of $t$. A word $w \in \Sigma^*$ is the lexicographic normal form of $\varphi(w)$ if and only if for each factor $aub$ of $w$ with $a, b \in \Sigma$, $u \in \Sigma^*$, satisfying $(au, b) \in I$, it holds $a <_\Sigma b$ [7, 53].

Let $x, y$ be vertices in the dependence graph of $t$ and let $lex(x, y)$ denote the predicate which means that $x$ is before $y$ in the lexicographic normal form. Assume that $lex(x, y)$ holds and let $x_0 \cdots x_n$ correspond to the factor in the lexicographic normal form with $x_0 = x$ and $x_n = y$. Let $i$ be minimal, $0 \leq i \leq n$, such that $x_i \leq y$ in the partial order. Then we have $(x_0 \cdots x_{i-1}, x_i) \in I$ and hence $\ell(x_0) \leq_\Sigma \ell(x_i)$. Therefore $lex(x, y)$ is equivalent with $\exists z\, (\ell(x) \leq_\Sigma \ell(z) \wedge z \leq y \wedge lex(x, z))$. This observation leads to a definition of $lex(x, y) = \bigvee_{a,b\in\Sigma} (P_a(x) \wedge P_b(y) \wedge lex_{a,b}(x, y))$ where $lex_{a,b}(x, y)$ is defined recursively as

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

The recursion ends after at most $2 \cdot |\Sigma|$ rounds and and yields a first-order formula (of exponential size in $|\Sigma|$). The quantifier depth is bounded by $|\Sigma| - 1$. $\square$

Note that for words, monadic second-order definability is equivalent to recognizability [33, 262], and first-order definability is equivalent to star-freeness [166], which is equivalent to aperiodicity [249]. The family of *star-free* finitary trace languages $\mathrm{SF}(\mathbb{M}(\Sigma, D))$ is the closure of the sets $\{a\}$ $(a \in \Sigma)$ by Boolean operations and concatenation and is equal to the set of aperiodic trace languages [129].

**Corollary 10.10.3** *A trace language $A \subseteq \mathbb{M}(\Sigma, D)$ is definable in monadic second-order logic $\mathcal{L}_2(\Sigma, D)$ if and only if it is recognizable.*

**Corollary 10.10.4** *A trace language $A \subseteq \mathbb{M}(\Sigma, D)$ is definable in first-order logic $\mathcal{L}_1(\Sigma, D)$ if and only if it is star-free.*

### 10.10.3 Equivalence of Recognizability and Monadic Second Order Logic for Finite and Infinite Traces

In the proof of the theorem above, the step from traces to words holds for any language of real traces. Since the lexicographic normal form is in general undefined for real traces, the step from words to traces can not be generalized directly to real traces. A different approach is needed to show the equivalence of recognizability and monadic second order logic definability for the family of real trace languages. We will use *c-rational trace languages* [205, 115]. The family of c-rational trace languages (Definition 11.7.31) is the closure of $\emptyset$ and the sets $\{a\}$ ($a \in \Sigma$) by concatenation, union, $*$-iteration and $\omega$-iteration, where iterations are restricted to connected c-rational trace languages (Corollary 11.7.24; for connectivity see Definition 6.3.5).

**Theorem 10.10.5** *Let $A \subseteq \mathbb{R}(\Sigma, D)$ be a real trace language. Then $A$ is recognizable if and only if it is definable in monadic second order logic $\mathcal{L}_2(\Sigma, D)$.*

**Proof:** The figure below sketches what we intend to show.

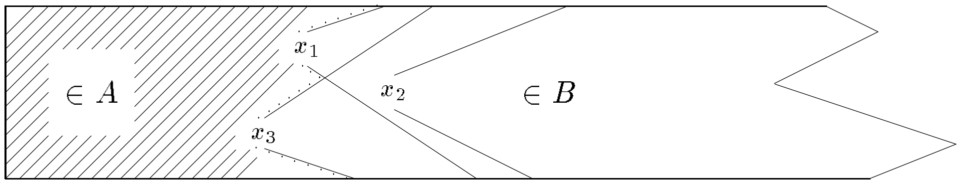
$$\begin{tikzcd}[column sep=small]
 & \text{$\varphi^{-1}(A)$ is definable in $\mathcal{L}_2(\Sigma)$} \arrow[r, Leftrightarrow, "\text{(Büchi)}"] & \text{$\varphi^{-1}(A)$ is recognizable} \arrow[dr, Leftrightarrow, "\text{Th. 11.7.4}"] & \\
\text{$A$ is definable in $\mathcal{L}_2(\Sigma, D)$} \arrow[ur, Rightarrow, "(1)"] & & & \text{$A$ is recognizable} \arrow[dl, Leftrightarrow, "\text{Th. 11.7.32}"] \\
 & & \text{$A$ is c-rational} \arrow[ull, Rightarrow, "(2)"] &
\end{tikzcd}$$

(1): See the the first part of the proof of Theorem 10.10.2.
(2): This implication is shown by induction over c-rational expressions. Since we deal simultaneously with finite and infinite traces, every formula states whether the satisfying traces are finite or not.

- Languages $\{a\}$ for $a \in \Sigma$: A formula stating that there is exactly one vertex, labelled by $a$.
- $A \cup B$ for c-rational sets $A$ and $B$: Combine the formulae $\psi_A$ and $\psi_B$ for $A$ and $B$ to $\psi_{A \cup B} = \psi_A \vee \psi_B$.
- $A \cdot B$ for c-rational sets $A \subseteq \mathbb{M}(\Sigma, D)$ and $B \subseteq \mathbb{R}(\Sigma, D)$: For $\psi_{A \cdot B}$ we need formulae with restricted quantification. Let $\psi|_X$ for some set variable $X$ be the formula $\psi$ where we replace every subformula $\exists x \psi'$ by $\exists x\ (x \in X \wedge \psi')$, $\forall x \psi'$ by $\forall x\ (x \in X \rightarrow \psi')$, $\exists Y \psi'$ by $\exists Y\ (Y \subseteq X \wedge \psi')$, and $\forall Y \psi'$ by $\forall Y\ (Y \subseteq X \rightarrow \psi')$. The formula $\psi_{A \cdot B}$ is now given by

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

  where the variables $x_1, \ldots, x_k$ cover the minimal elements of the suffix claimed to be in $B$.




- $A^*$, $A^\omega$ for a connected c-rational set $A \subseteq \mathbb{M}(\Sigma, D)$: Let $\psi_A$ denote the formula defining $A$ and assume for convenience $1 \notin A$.

  The formula below is based on a colouring of the vertices of the dependence graph of a factorization $t_1 t_2 \ldots$, such that every factor $t_i \in \mathbb{M}(\Sigma, D)$ is one-coloured and, if two different factors have the same colour, then there is no edge between them in the Hasse diagram. Two factors $t_i, t_j$ will have the same colour only if they have the same alphabet, $\mathrm{alph}(t_i) = \mathrm{alph}(t_j)$. Hence, for every $\Sigma' \subseteq \Sigma$ we take two colours and colour alternatingly the factors $t_i$ with the colours of $\Sigma' = \mathrm{alph}(t_i)$. (Note that factors with the same alphabet are totally ordered.)

  We define $\psi_{A^*}$ and $\psi_{A^\omega}$ as

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

  where $X_1, \ldots, X_k$ express the colouring mentioned above, with $k = 2 \cdot 2^{|\Sigma|}$.

  Let $\psi_1$ be a formula stating that $V$ is the disjoint union of all $X_i$. For the next two subformulae we define some abbreviations. By $mocs(X)$ we express that $X$ is a "maximal one-coloured connected subgraph" of the Hasse diagram; formally, we define $mocs(X)$ as

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

  where $E$ is the edge relation of the Hasse diagram. The subformula "$X$ is connected" simply states that for every $x, y \in X$ there is an undirected path between $x$ and $y$, lying in $X$.

  The formula $\psi_2$ ensures that every $mocs$-component is an element of $A$ and is defined as

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

  Note that since $A$ is a finitary language, the $mocs$-components satisfying $\psi_2$ will be finite.

The underlying interpretation of the *mocs*-components is being factors of the given trace which belong to $A$. In order to ensure that the *mocs*-components can be ordered, thus corresponding to a factorization, we first define $X \prec Y$ as

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

The formula $\psi_3$ ensures that there are no cycles in the relation $\prec$ restricted to *mocs*-components. Note that cycles longer than $|\Sigma|$ can be shortened. Therefore we define $\psi_3$ as follows:

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

Finally $\psi_4$ expresses whether the trace is finite or not, depending on the type of iteration ($*$ or $\omega$).

□

**Remark 10.10.6** In Theorem 10.10.2 we directly transform formulae on finite words to formulae on finite traces and vice versa. Note that this provides, together with Theorem 10.10.5, a new proof for Métivier's [195, Theorem 2.3] and Ochmański's [205, Lemma 8.2] result on the recognizability of the iteration of connected recognizable languages of finite traces (Prop. 6.3.11).

**Remark 10.10.7** Theorem 10.10.5 can also be proved [90] using Büchi asynchronous (cellular) automata, first investigated by P. Gastin and A. Petit [113].

### 10.10.4 Equivalence of First Order Logic and Star-Free Expressions for Finite and Infinite Traces

We already introduced the family of *star-free* finitary trace languages $\mathrm{SF}(\mathbb{M}(\Sigma, D))$ as the closure of singleton sets $\{a\}$ ($a \in \Sigma$) by Boolean operations and concatenation. The family $\mathrm{SF}(\mathbb{R}(\Sigma, D))$ of *star-free real trace languages* is the closure of $\mathrm{SF}(\mathbb{M}(\Sigma, D))$ by concatenation from the left with languages of $\mathrm{SF}(\mathbb{M}(\Sigma, D))$ and by Boolean operations, where the complement is taken with respect to $\mathbb{R}(\Sigma, D)$.

We will consider formulae and languages with *free variables*. For a finite set $W$ of variables we add to the structure $(V, <, \ell)$ a mapping $\sigma : W \to V$, associating every variable to a vertex in the dependence graph of the trace. Thus we obtain a new structure $(V, <, \ell, \sigma)$ of a trace with variables. If a trace has no variables, then the new structure $(V, <, \ell, \emptyset)$ can be identified with $(V, <, \ell)$. The sets $\mathbb{M}(\Sigma, D)$, $\mathbb{R}(\Sigma, D)$, $A$, $B$, ... with variables $W$ are denoted by $\mathbb{M}_W$, $\mathbb{R}_W$, $A_W$, $B_W$, ...; a trace $t$ with variable set $W$ is denoted by $t_W$. Often we omit the subscript of a set $A_W$; we write $W_t$ for the set of variables of $t \in \mathbb{R}_W$. Concatenation of traces with variables is defined only on traces with disjoint sets of variables. If $u = (V_u, <_u, \ell_u, \sigma_u)$ and $v = (V_v, <_v, \ell_v, \sigma_v)$ are traces with sets $W_u$, $W_v$ of variables and $W_u \cap W_v = \emptyset$, then the dependence graph of the concatenation $t = uv = (V_t, <_t, \ell_t, \sigma_t)$ is defined as the usual concatenation of the two traces together with the mapping $\sigma_t = \sigma_u \cup \sigma_v$. The complement of a language $A_W \subseteq \mathbb{R}_W$ is taken with respect to $\mathbb{R}_W$.

The following lemma states some useful properties of star-free trace languages over the extended structures defined above. We define the left and right quotients $t^{-1}A$ and $At^{-1}$ for $t \in \mathbb{R}_U$, $A \subseteq \mathbb{R}_W$ as $t^{-1}A = \{\, v \mid tv \in A \,\}$ and $At^{-1} = \{\, v \mid vt \in A \,\}$. Note that the partial monoid of real traces $\mathbb{R}$ is not 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 10.10.8** *1. For any subalphabet $\Sigma' \subseteq \Sigma$ and finite variable set $W$, the languages $\{t \in \mathbb{M}_W \mid \mathrm{alph}(t) \subseteq \Sigma'\}$ and $\{t \in \mathbb{R}_W \mid \mathrm{alph}(t) \subseteq \Sigma'\}$ are star-free.*

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

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

**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: Let $A \in \mathrm{SF}(\mathbb{R}_W)$. 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 $A = \{a\}$ for some $a \in \Sigma_W$, then $t^{-1}A = \{t^{-1}a\}$ and $n_A \leq 2$.

If $A = B \cup C$, then $\{\, t^{-1}(B \cup C) \mid t \in \mathbb{R}_U \,\} = \{\, t^{-1}B \cup t^{-1}C \mid t \in \mathbb{R}_U \,\} \subseteq \{\, t^{-1}B \cup t'^{-1}C \mid t, t' \in \mathbb{R}_U \,\}$ and $n_A \leq n_B \cdot n_C$ ($A = B \cap C$ analogously).

If $A = \overline{B}$, then with $t^{-1}\overline{B} = \overline{t^{-1}B} \setminus \{v \in \mathbb{R}_{W\setminus U} \mid D \cap \mathrm{alph}(v) \times \mathrm{alphinf}(t) \neq \emptyset\}$ we obtain $n_A \leq n_B \cdot 2^{|\Sigma|}$. (By $\mathrm{alphinf}(t)$ we denote the set of letters occuring infinitely often in $t$.)

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

$$\{ t^{-1}(BC) \mid t \in \mathbb{R}_U \} = \{ \bigcup_{rs=t} \underbrace{(\underbrace{r^{-1}B}_{\leq n_B} \cap \underbrace{F_s}_{\leq 2^{|\Sigma|}})}_{\leq n_B \cdot 2^{|\Sigma|}} \underbrace{(s^{-1}C)}_{\leq n_C} \mid t \in \mathbb{R}_U \} \ ,$$

and $n_A \leq 2^{n_B\, 2^{|\Sigma|}\, n_C}$.
Right quotients are treated symmetrically.

□

**Theorem 10.10.9** *A trace language $A \subseteq \mathbb{R}(\Sigma, D)$ is definable in first order logic $\mathcal{L}_1(\Sigma, D)$ if and only if it is star-free.*

**Proof:** "⇐": In order to construct logical formulae from star-free expressions, we replace the set operations by the corresponding logical operations. For concatenation we use the first order formula given in Theorem 10.10.5.

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

*Predicates:* The set of infinite traces with a finite set of variables $W$ satisfying $x < y$ for some variables $x, y \in W$, is the star-free trace language (we omit the subscripts for $\mathbb{M}, \mathbb{R}$)

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

Note that the variable sets omitted above form a partition of $W$ and $l \leq |\Sigma|$.

The language with set of variables $W$ satisfying $P_a(x)$ is $\bigcup_{\text{finite}} \ \mathbb{M}a\mathbb{R}$ with $x \in W_a$. For $x = y$ we have $\bigcup_{\text{finite}} \bigcup_{\substack{a \in \Sigma_W \\ x,y \in W}} \mathbb{M}a\mathbb{R}$.

*Induction step:* For $\wedge$, $\vee$, $\neg$ we use the corresponding set theoretical operations. Finally, for quantified formulae, if $A \in \mathrm{SF}(\mathbb{R})$ is a language defined by $\psi$ and $x \in W_A$, then we can represent $A$ as [224]

$$A = \bigcup_{\substack{a \in \Sigma,\ u\,a \in \mathbb{R} \\ x \in W_a}} B(u)\, a\, C(u)$$

with

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

By Lemma 10.10.8 the union in the expression for $A$ is finite. Finally, the language defined by the formula $\exists x \psi$ is

$$A' = \bigcup_{u\,a \in \mathbb{R}} B(u)\, a\, C(u)$$

with $W_{A'} = W_A \setminus \{x\}$. □

Similar to the result of Thomas [261] and of Perrin and Pin [224], this can be generalized to embed a hierarchy of first order formulae, where the alternations of negation and existential quantification are counted, between two hierarchies of star-free languages, where the alternations of complementation and concatenation are counted [90].

### 10.10.5 Notes

Monadic second order logic on traces has been first investigated by Thomas [263] using an automata-theoretic approach [262, for example]. These results have been generalized to infinite traces and extended to the first order case [90, 91]. Independently, the first order case has also been considered by Thomas and Zielonka (personal communication).

In the present appendix we have used single traces as underlying structures instead of trace structures as in the other parts of this chapter. This yields the decidability of the logic as an immediate consequence of decidability in the word case, shown by Büchi [33, 262]. This appendix does not deal with temporal logic. A temporal logic based on single traces as underlying structures has been proposed in [90]. As a main result about this logic it is shown there that for finite traces definability in this new logic is equivalent with definability in first order logic.