Automata, Languages and Programming, 29th International Colloquium, Proceedings:
P. Widmayer, F.T. Ruiz, R. Morales, M. Hennessy, S. Eidenbenz, R. Conejo(eds.)
Springer Lecture Notes in Computer Science **2382** (2002), 938–949.

An Elementary Expressively Complete Temporal Logic for Mazurkiewicz Traces*

Paul Gastin[1] and Madhavan Mukund[2]

[1] LIAFA, Université Paris 7, 2, place Jussieu, F-75251 Paris Cedex 05
`Fax: 33 1 44 27 68 49` — `Paul.Gastin@liafa.jussieu.fr`
[2] Chennai Mathematical Institute, 92 G N Chetty Road, Chennai 600 017, India
`Fax: 91 44 823 3671` — `madhavan@cmi.ac.in`

Abstract. In contrast to the classical setting of sequences, no temporal logic has yet been identified over Mazurkiewicz traces that is equivalent to first-order logic over traces and yet admits an elementary decision procedure. In this paper, we describe a local temporal logic over traces that is expressively complete and whose satisfiability problem is in PSPACE. Contrary to the situation for sequences, past modalities are essential for such a logic. A somewhat unexpected corollary is that first-order logic with three variables is expressively complete for traces.
Keywords Temporal logics, Mazurkiewicz traces, concurrency

1 Introduction

Linear-time temporal logic (LTL) [15] has established itself as a useful formalism for specifying the interleaved behaviour of reactive systems. To combat the combinatorial blow-up involved in describing computations of concurrent systems in terms of interleavings, there has been a lot of interest in using temporal logic more directly on labelled partial orders.

Mazurkiewicz traces [12] are labelled partial orders generated by dependence alphabets of the form (Σ, D), where D is a *dependence* relation over Σ. If $(a, b) \notin D$, a and b are deemed to be independent actions that may occur concurrently. Traces are a natural formalism for describing the behaviour of static networks of communicating finite-state agents [26].

LTL over Σ-labelled sequences is equivalent to $\mathrm{FO}_\Sigma(<)$, the first-order logic over Σ-labelled linear orders [11] and thus defines the class of aperiodic languages over Σ. Though $\mathrm{FO}_\Sigma(<)$ permits assertions about both the past and the future, future modalities suffice for establishing the expressive completeness of LTL with respect to $\mathrm{FO}_\Sigma(<)$ [9]. From a practical point of view, a finite-state program may be checked against an LTL specification relatively efficiently [2, 19, 20].

Though a number of LTL-like temporal logics have been proposed for reasoning directly about Mazurkiewicz traces, it has proved elusive to define a tractable logic that is expressively complete with respect to $\mathrm{FO}_\Sigma(<)$, the first order logic

* Work done while the second author was visiting LIAFA, Université Paris 7. Partial support of CEFIPRA-IFCPAR Project 2102-1 (ACSMV) is gratefully acknowledged.

over traces. The first expressively complete temporal logic over traces was described in [17]. The result was refined in [3, 5] to show expressive completeness without past modalities, using an extension of the proof technique developed for LTL in [24, 25]. Formulae in both these logics are defined at *global configurations* (maximal antichains). Unfortunately, reasoning at the level of global configurations makes the complexity of deciding satisfiability non-elementary [21].

Computational tractability seems to require interpreting formulae at *local states*—effectively at individual events. This approach was followed (sometimes indirectly) in early temporal logics over traces [1, 13, 16, 18]. While model-checking is relatively efficient for these logics, they are not known to be expressively complete. In fact, the logic TLC described in [1] is not even first-order definable.

Recently, an expressively complete local temporal logic has been defined for the subclass of Mazurkiewicz traces where the dependence graph of the underlying alphabet has a series-parallel structure [4]. Unfortunately, this logic *cannot* be expressively complete over all traces—the logic uses only future modalities and, unlike LTL over sequences, there exist two first-order inequivalent traces that cannot be distinguished using only future modalities [23].

In this paper, we define a new local temporal logic over traces that is expressively complete and whose satisfiability problem is in PSPACE. The observation in [23] about the shortcoming of future modalities necessitates the introduction of past modalities in our logic. As in [3, 4], we show expressive completeness using an extension to traces of the proof technique introduced in [24, 25] for LTL over sequences. However, the earlier proofs exploit the fact that the past cannot affect the truth of a formula. The introduction of past modalities in our logic requires the use of new techniques for "relativizing" formulae placed in new past contexts. The satisfiability problem for our logic is settled using two-way alternating automata, extending techniques from [1, 4].

A corollary of our main result is that $\mathrm{FO}^3_{(\Sigma,D)}(<)$, the subclass of first-order logic formulae with at most three variables, is expressively complete over traces, extending the same result for sequences. This is somewhat surprising—intuition suggests that we require variables proportional to the width of a trace. (This result has been independently established by Walukiewicz [22].)

The paper is organized as follows. We begin with some preliminaries about traces. In Section 3 we define our new temporal logic. Section 4 describes a syntactic partition of traces that is used in Section 5 to establish expressive completeness. The decision procedure for satisfiability is sketched in Section 6. We conclude with a discussion of open questions. Many proofs have had to be omitted in this extended abstract.

2 Preliminaries

We briefly recall some notions about Mazurkiewicz traces (see [6] for background). A *dependence alphabet* is a pair (Σ, D) where the alphabet Σ is a finite set of actions and the *dependence relation* $D \subseteq \Sigma \times \Sigma$ is reflexive and symmetric. The *independence relation* I is the complement of D. For $A \subseteq \Sigma$, the set of letters independent of A is denoted by $I(A) = \{b \in \Sigma \mid (a,b) \in I \text{ for all } a \in A\}$ and the set of letters depending on (some action in) A is denoted by $D(A) = \Sigma \setminus I(A)$.

A *Mazurkiewicz trace* is a labelled partial order $t = [V, \leq, \lambda]$ where V is a set of vertices labelled by $\lambda : V \to \Sigma$ and $\leq$ is a partial order over V satisfying the following conditions: For all $x \in V$, the downward set $\downarrow x = \{y \in V \mid y \leq x\}$ is finite, $(\lambda(x), \lambda(y)) \in D$ implies $x \leq y$ or $y \leq x$, and $x \lessdot y$ implies $(\lambda(x), \lambda(y)) \in D$, where $\lessdot \; = \; < \setminus <^2$ is the immediate successor relation in t.

The *alphabet* of a trace t is the set $\mathrm{alph}(t) = \lambda(V) \subseteq \Sigma$ and its *alphabet at infinity*, $\mathrm{alphinf}(t)$, is the set of letters occurring infinitely often in t. The set of all traces is denoted by $\mathbb{R}(\Sigma, D)$ or simply by $\mathbb{R}$. A trace t is called *finite* if V is finite. For $t = [V, \leq, \lambda] \in \mathbb{R}$, we define $\min(t) \subseteq V$ as the set of all minimal vertices of t. We can also read $\min(t) \subseteq \Sigma$ as the set of labels of the minimal vertices of t. It will be clear from the context what we actually mean.

Let $t_i = [V_i, \leq_i, \lambda_i]$, $i \in \{1, 2\}$, be a pair of traces such that $V_1 \cap V_2 = \emptyset$ and $\mathrm{alphinf}(t_1) \times \mathrm{alph}(t_2) \subseteq I$. We define the concatenation of t_1 and t_2 to be $t_1 \cdot t_2 = [V, \leq, \lambda]$ where $V = V_1 \cup V_2$, $\lambda = \lambda_1 \cup \lambda_2$ and $\leq$ is the transitive closure of $\leq_1 \cup \leq_2 \cup (V_1 \times V_2 \cap \lambda^{-1}(D))$. The set of finite traces is then a monoid, denoted $\mathbb{M}(\Sigma, D)$ or simply $\mathbb{M}$, with the empty trace $1 = (\emptyset, \emptyset, \emptyset)$ as unit.

Here is some useful notation for subclasses of traces. For $C \subseteq \Sigma$, let $\mathbb{R}_C = \{t \in \mathbb{R} \mid \mathrm{alph}(t) \subseteq C\}$ and $\mathbb{M}_C = \mathbb{M} \cap \mathbb{R}_C$. Also, $(\mathrm{alphinf} = C) = \{t \in \mathbb{R} \mid \mathrm{alphinf}(t) = C\}$ and $(\min = C) = \{t \in \mathbb{R} \mid \min(t) = C\}$. For $A, C \subseteq \Sigma$, we set $\mathbb{R}_C^A = \mathbb{R}_C \cap (\mathrm{alphinf} = A)$. Observe that $\mathbb{M}_C = \mathbb{R}_C^{\emptyset}$.

The first order logic over traces $\mathrm{FO}_\Sigma(<)$ is given by the syntax:

$$\varphi ::= P_a(x) \mid x < y \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x \varphi,$$

where $a \in \Sigma$ and $x, y \in \mathrm{Var}$ are first order variables. For a trace $t = [V, \leq, \lambda]$ and a valuation $\sigma : \mathrm{Var} \to V$, $t, \sigma \models \varphi$ denotes that t satisfies φ under σ. We interpret each predicate P_a by the set $\{x \in V \mid \lambda(x) = a\}$ and the relation $<$ as the strict partial order relation of t. The semantics then lifts to all formulas as usual. Since the meaning of a closed formula (sentence) φ is independent of the valuation σ, we can associate with each sentence φ the language $\mathcal{L}(\varphi) = \{t \in \mathbb{R} \mid t \models \varphi\}$. A trace language $L \subseteq \mathbb{R}$ is said to be expressible in $\mathrm{FO}_\Sigma(<)$ if there is a sentence $\varphi \in \mathrm{FO}_\Sigma(<)$ such that $L = \mathcal{L}(\varphi)$. For $n > 0$, $\mathrm{FO}_\Sigma^n(<)$ denotes the set of formulae with at most n distinct variables (possibly bound and reused several times).

We use the algebraic notion of recognizability (see e.g. [14]). Let $h : \mathbb{M} \to S$ be a morphism to a finite monoid S. For $t, u \in \mathbb{R}$, we say that t and u are h-similar, denoted $t \sim_h u$, if either $t, u \in \mathbb{M}$ and $h(t) = h(u)$ or t and u have infinite factorizations in non-empty finite traces $t = t_1 t_2 \cdots$, $u = u_1 u_2 \cdots$ with $h(t_i) = h(u_i)$ for all i. The transitive closure $\approx_h$ of $\sim_h$ is an equivalence relation. Since S is finite, this equivalence relation is of finite index with at most $|S|^2 + |S|$ equivalence classes. A trace language $L \subseteq \mathbb{R}$ is *recognized* by h if it is saturated by $\approx_h$ (or equivalently by $\sim_h$), i.e., $t \in L$ implies $[t]_{\approx_h} \subseteq L$ for all $t \in \mathbb{R}$.

Let $L \subseteq \mathbb{R}$ be recognized by a morphism $h : \mathbb{M} \to S$. For $B \subseteq \Sigma$, $L \cap \mathbb{M}_B$ and $L \cap \mathbb{R}_B$ are recognized by $h\restriction_{\mathbb{M}_B}$ the restriction of h to $\mathbb{M}_B$.

A finite monoid S is *aperiodic* if there is an $n \geq 0$ such that $s^n = s^{n+1}$ for all $s \in S$. A trace language $L \subseteq \mathbb{R}$ is *aperiodic* if it is recognized by some morphism to a finite and aperiodic monoid.

Theorem 1 ([8, 7]). *A language $L \subseteq \mathbb{R}(\Sigma, D)$ is expressible in $\mathrm{FO}_\Sigma(<)$ if and only if it is aperiodic.*

3 Local temporal logic

Our local temporal logic over traces consists of two types of formulae: *internal formulae*, evaluated at arbitrary events within a trace, and *initial formulae*, asserting properties of minimal events of the trace. Internal formulae only talk about the structure within a connected portion of a trace. Initial formulae permit us to combine local assertions across disjoint components and form "global" assertions about the structure of the trace.

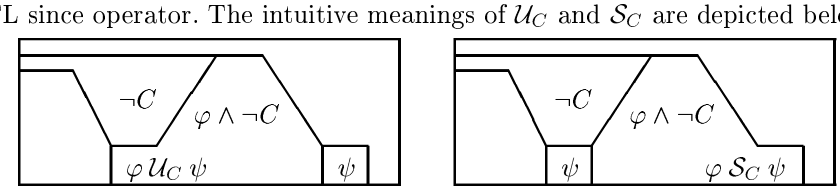
The set $\mathrm{LocTL}^{\mathrm{i}}_\Sigma$ of internal formulae over the alphabet Σ is defined as follows:

$$\varphi ::= a \in \Sigma \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \varphi\,\mathcal{U}_C\,\varphi, C \subseteq \Sigma \mid \mathsf{EY}\,\varphi \mid \varphi\,\mathcal{S}_C\,\varphi, C \subseteq \Sigma$$

Let $t = [V, \leq, \lambda] \in \mathbb{R}$ be a finite or infinite trace and let $x \in V$ be some vertex of t. We write $t, x \models \varphi$ to denote that trace t at node x satisfies the formula $\varphi \in \mathrm{LocTL}^{\mathrm{i}}_\Sigma$. This is defined inductively as follows:

$$\begin{array}{lll}
t, x \models a & \text{if} & \lambda(x) = a \\
t, x \models \neg\varphi & \text{if} & t, x \not\models \varphi \\
t, x \models \varphi \vee \psi & \text{if} & t, x \models \varphi \text{ or } t, x \models \psi \\
t, x \models \mathsf{EX}\,\varphi & \text{if} & \exists y.\ x \lessdot y \text{ and } t, y \models \varphi \\
t, x \models \mathsf{EY}\,\varphi & \text{if} & \exists y.\ y \lessdot x \text{ and } t, y \models \varphi \\
t, x \models \varphi\,\mathcal{U}_C\,\psi & \text{if} & \exists z \geq x.\ [t, z \models \psi \text{ and } \forall y.\ (x \leq y < z) \Rightarrow t, y \models \varphi \text{ and} \\
 & & \qquad \forall y.\ (\lambda(y) \in C \text{ and } y \leq z) \Rightarrow y \leq x] \\
t, x \models \varphi\,\mathcal{S}_C\,\psi & \text{if} & \exists z \leq x.\ [t, z \models \psi \text{ and } \forall y.\ (z < y \leq x) \Rightarrow t, y \models \varphi \text{ and} \\
 & & \qquad \forall y.\ (\lambda(y) \in C \text{ and } y \leq x) \Rightarrow y \leq z]
\end{array}$$

The modalities EX and EY are existential extensions of the next state and previous state operators X and Y of LTL. The modality $\mathcal{U}_C$ extends the "universal" until operator $\mathcal{U}$ defined in [4] with an alphabetic *filter* that restricts the actions performed while fulfilling an until requirement. If $t, x \models \varphi\,\mathcal{U}_C\,\psi$ then no C-actions are performed when going from x to the node satisfying ψ. Since the filter also applies to events that are concurrent to x, the modality $\mathcal{U}_C$ no longer looks purely at the future of an event. Similarly, $\mathcal{S}_C$ is a filtered extension of the LTL since operator. The intuitive meanings of $\mathcal{U}_C$ and $\mathcal{S}_C$ are depicted below.



Past modalities are essential, as indicated by the following example from [23], where the dependence relation is $a - b - c - d$. These two traces are not first-order equivalent but are bisimilar at the level of events and thus cannot be distinguished by pure future modalities. However, the two traces below can be distinguished using the filtered until, which is not pure future.

$$\begin{array}{ccc} a \to b \to c \to b \to c \cdots & \qquad & d \to c \to b \to c \to b \cdots \\ \quad\uparrow & & \quad\uparrow \\ d \to c & & a \to b \end{array}$$

For the same alphabet, one can also show that the language $ad(bc)^* \subseteq \mathbb{M}$ of finite traces is first order but cannot be expressed by pure future modalities.

As usual, we can derive useful operators. Using unfiltered versions of until and since, $\varphi \mathcal{U} \psi = \varphi \mathcal{U}_\emptyset \psi$ and $\varphi \mathcal{S} \psi = \varphi \mathcal{S}_\emptyset \psi$, we recover the classical modalities *eventually in the future* $\mathsf{F}\,\varphi = \top\,\mathcal{U}\,\varphi$, *eventually in the past* $\mathsf{F}^{-1}\,\varphi = \top\,\mathcal{S}\,\varphi$, *always in the future* $\mathsf{G}\,\varphi = \neg\,\mathsf{F}\,\neg\varphi$, and *always in the past* $\mathsf{G}^{-1}\,\varphi = \neg\,\mathsf{F}^{-1}\,\neg\varphi$. We also use some filtered versions of these modalities: $\mathsf{F}_b\,\varphi = \top\,\mathcal{U}_b\,\varphi$ (eventually in the future but above the same b's), $\mathsf{G}_b\,\varphi = \neg\,\mathsf{F}_b\,\neg\varphi$ (for all future vertices that are above the same b's). The existence of a successor (resp. a predecessor) above the same b's can be written as follows:

$$\begin{aligned} \mathsf{EX}_b\,\varphi &= \bigvee_{d \neq b} \mathsf{EX}(d \wedge \varphi) \wedge \top\,\mathcal{U}_b\,d = \bigvee_{c} c \wedge \mathsf{EX}(\neg b \wedge \varphi \wedge \top\,\mathcal{S}_b\,c), \\ \mathsf{EY}_b\,\varphi &= \bigvee_{c} \mathsf{EY}(c \wedge \varphi) \wedge \top\,\mathcal{S}_b\,c \ = \bigvee_{d \neq b} d \wedge \mathsf{EY}(\varphi \wedge \top\,\mathcal{U}_b\,d). \end{aligned}$$

Finally, $\mathsf{F}^{\infty}\,a = \mathsf{F}\,a \wedge \mathsf{G}(a \Rightarrow \mathsf{EX}\,\mathsf{F}\,a)$ expresses the existence of infinitely many vertices labelled with a above the current vertex. The filtered version of this formula, $\mathsf{F}^{\infty}_b\,a = \mathsf{F}_b\,a \wedge \mathsf{G}_b(a \Rightarrow \mathsf{EX}_b\,\mathsf{F}_b\,a)$, requires in addition that the vertices labelled with a are above the same b's.

For $A \subseteq \Sigma$, let A denote the internal formula $\bigvee_{a \in A} a$. An *alphabetic since* is a formula of the form $A\,\mathcal{S}_C\,a$ for $A, C \subseteq \Sigma$ and $a \in \Sigma$.

We turn now to initial formulae. These are boolean combinations of formulae $\mathsf{EM}\,\varphi$ asserting that some minimal vertex in the trace satisfies the internal formula φ. Formally, the set LocTL_Σ of initial formulae over Σ is given by:

$$\alpha ::= \bot \mid \mathsf{EM}\,\varphi, \varphi \in \mathrm{LocTL}^{\mathrm{i}}_\Sigma \mid \neg\alpha \mid \alpha \vee \alpha$$

The semantics is defined as follows:

$$\begin{array}{lll} t \models \mathsf{EM}\,\varphi & \text{if} & \exists x.\ (x \in \min(t) \text{ and } t, x \models \varphi) \\ t \models \neg\alpha & \text{if} & t \not\models \alpha \\ t \models \alpha \vee \beta & \text{if} & t \models \alpha \text{ or } t \models \beta \end{array}$$

An initial formula $\alpha \in \mathrm{LocTL}_\Sigma$ defines the trace language $\mathcal{L}(\alpha) = \{t \in \mathbb{R} \mid t \models \alpha\}$. We can then express various alphabetic properties using initial formulae: $\mathcal{L}(\mathsf{EM}\,a) = \{t \in \mathbb{R} \mid a \in \min(t)\}$, $\mathcal{L}(\mathsf{EM}\,\mathsf{F}\,a) = \{t \in \mathbb{R} \mid a \in \mathrm{alph}(t)\}$, and $\mathcal{L}(\mathsf{EM}\,\mathsf{F}^{\infty}\,a) = \{t \in \mathbb{R} \mid a \in \mathrm{alphinf}(t)\}$. Therefore, for $C \subseteq \Sigma$, trace languages such as $(\mathrm{alphinf} = C)$ and $(\min = C)$ are expressible in LocTL_Σ.

The following result is immediate from the definition of LocTL_Σ.

Proposition 2. *If a trace language is expressible in* LocTL_Σ*, then it is expressible in* $\mathrm{FO}^3_\Sigma(<)$*.*

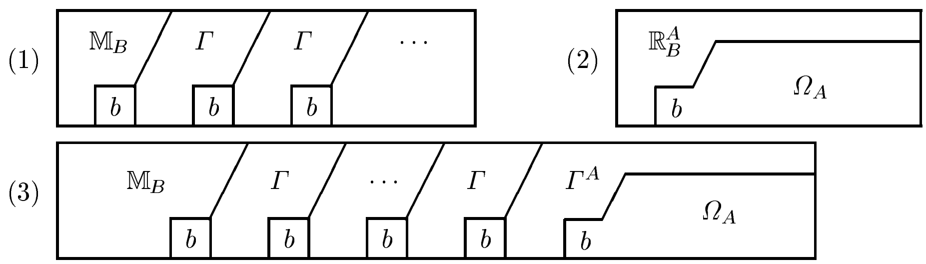
4 Decomposition of traces

The proof of our main result is a case analysis based on partitioning the set of traces according to the structure of the trace. Fix a letter $b \in \Sigma$ and set $B = \Sigma \setminus \{b\}$. Using the notation introduced in the middle of Section 2, let $\Gamma^A = \{t \in \mathbb{R}^A_B \mid \min(t) \subseteq D(b)\}$, $\Gamma = \Gamma^\emptyset$, and $\Omega_A = \{t \in \mathbb{R}_{I(A)} \mid \min(t) \subseteq \{b\}\}$.

Each trace $t \in \mathbb{R}$ has a unique factorization $t = t_0 b t_1 b t_2 \cdots$ with $t_0 \in \mathbb{R}_B$ and $t_i \in \mathbb{R}_B \cap (\min \subseteq D(b))$ for all $i > 0$. We obtain the following partition of $\mathbb{R}$.

$$\mathbb{R} = \mathbb{M}_B\,(b\Gamma)^\infty \uplus \left(\bigcup_{A\neq\emptyset} \mathbb{R}^A_B\,\Omega_A\right) \uplus \left(\bigcup_{A\neq\emptyset} \mathbb{M}_B\,(b\Gamma)^* b\Gamma^A\,\Omega_A\right)$$

If all t_i are finite, then we are in the first block. If $t_0 \notin \mathbb{M}$, then we are in the second block. Otherwise, we are in the last block. Note that $\mathbb{M}_B\,(b\Gamma)^* = \mathbb{M}$. The three possibilities are depicted below.



The following two results will allow us to use this decomposition effectively in proving the expressive completeness of our logic.

Lemma 3. *All these sets are expressible in* LocTL_Σ*. More precisely,*

1. *The sets* $\mathbb{M}_B\,(b\Gamma)^*$ *and* $\mathbb{M}_B\,(b\Gamma)^\omega$ *are expressible in* LocTL_Σ*.*
2. $\mathbb{R}^A_B \cdot \Omega_A = \mathcal{L}(\alpha)$ *where*

$$\alpha = \left(\bigwedge_{a\in A} \mathsf{EM}(\neg b \wedge \mathsf{F}^\infty_b\, a)\right) \wedge \left(\bigwedge_{a\notin A} \neg\,\mathsf{EM}(\neg b \wedge \mathsf{F}^\infty_b\, a)\right).$$

3. $\mathbb{M}_B\,(b\Gamma)^* b\Gamma^A\,\Omega_A = \mathbb{M}b\Gamma^A\,\Omega_A = \mathcal{L}(\beta)$ *where*

$$\beta = \bigvee_{C\subseteq\Sigma}\left((\mathrm{alphinf} = C) \wedge \mathsf{EM}\,\mathsf{F}\left(b \wedge \bigwedge_{c\in C} \mathsf{F}^\infty\, c \wedge \bigwedge_{a\in A} \mathsf{F}^\infty_b\, a \wedge \bigwedge_{a\notin A} \neg\,\mathsf{F}^\infty_b\, a\right)\right).$$

Note that "the" b *in* $\mathbb{M}b\Gamma^A\,\Omega_A$ *is characterized by the formula* $b \wedge \mathsf{F}^\infty_b\, a$*, where* a *is any letter in* A*.*

Lemma 4. *Let $A \subseteq \Sigma$ and let $L \subseteq \mathbb{R}$ be a trace language recognized by a morphism h from $\mathbb{M}$ into a finite monoid S. Then,*

$$
\begin{aligned}
(1)\quad L \cap \mathbb{M}_B (b\Gamma)^\infty &= \bigcup_{\textit{finite}} (L_1 \cap \mathbb{M}_B) \cdot (L_2 \cap (b\Gamma)^\infty)\\
(2)\quad L \cap \mathbb{R}_B^A \,\Omega_A &= \bigcup_{\textit{finite}} (L_1 \cap \mathbb{R}_B^A) \cdot (L_2 \cap \Omega_A)\\
(3)\quad L \cap \mathbb{M} b \Gamma^A \,\Omega_A &= \bigcup_{\textit{finite}} (L_1 \cap \mathbb{M}_B)(L_2 \cap (b\Gamma)^*) b (L_3 \cap \Gamma^A)(L_4 \cap \Omega_A)
\end{aligned}
$$

where the trace languages $L_i \subseteq \mathbb{R}$ are recognized by h.

5 Expressive completeness

We now show that our temporal logic for traces is expressively complete with respect to $\mathrm{FO}_\Sigma(<)$. In light of Theorem 1, it suffices to establish the following.

Theorem 5. *Every aperiodic trace language $L \subseteq \mathbb{R}$ is expressible in* LocTL_Σ.

From Proposition 2 and Theorems 1 and 5, we deduce immediately

Corollary 6. *Let $L \subseteq \mathbb{R}$ be a trace language. The following are equivalent*

1. L is aperiodic,
2. L is expressible in LocTL_Σ,
3. L is expressible in $\mathrm{FO}^3_\Sigma(<)$,
4. L is expressible in $\mathrm{FO}_\Sigma(<)$.

In the rest of this section, we sketch a proof of Theorem 5. We start with a trace language $L \subseteq \mathbb{R}$ recognized by a morphism h from $\mathbb{M}$ to a finite aperiodic monoid S. We use induction on $|\Sigma|$.

The base case is when $\Sigma = \{a\}$, a singleton. Then L is either a finite set or the union of a finite set and a set of the form $a^n a^*$, $n \geq 0$. In both cases, L is expressible in LocTL_Σ. For instance, $\{a^\omega\}$ and $\{a^2\}$ correspond to the formulae $\mathsf{EM}\,\mathsf{G}\,\mathsf{EX}\,\top$ and $\mathsf{EM}\,\mathsf{EX}\,\neg\,\mathsf{EX}\,\top$. Also, $a^3a^* \cup \{a^\omega\}$ is expressed by $\mathsf{EM}\,\mathsf{EX}\,\mathsf{EX}\,\top$.

For the induction step, with $|\Sigma| > 1$, we fix a letter $b \in \Sigma$ and use the decomposition introduced in Section 4. Therefore, we have

$$L = \Big(L \cap \mathbb{M}_B (b\Gamma)^\infty\Big) \uplus \Big(L \cap \bigcup_{A \neq \emptyset} \mathbb{R}_B^A \,\Omega_A\Big) \uplus \Big(L \cap \bigcup_{A \neq \emptyset} \mathbb{M}_B (b\Gamma)^* b \Gamma^A \,\Omega_A\Big).$$

We show separately that the three languages above are expressible in LocTL_Σ. From Lemma 4, it suffices to establish the following.

Proposition 7. *Recall that $B = \Sigma \setminus \{b\}$. Let $A \subseteq \Sigma$ be non-empty and let $L \subseteq \mathbb{R}$ be a trace language recognized by h. Then,*

(1) $(L \cap \mathbb{R}_B) \cdot (\min \subseteq \{b\})$ *is expressible in* LocTL_Σ.
(2) $\mathbb{M}_B \cdot (L \cap (b\Gamma)^\infty)$ *is expressible in* LocTL_Σ.
(3) $L \cap \mathbb{M}_B (b\Gamma)^\infty$ *is expressible in* LocTL_Σ.

(4) $\mathbb{R}_B^A \cdot (L \cap \Omega_A)$ *is expressible in* LocTL_Σ.
(5) $L \cap \mathbb{R}_B^A \Omega_A$ *is expressible in* LocTL_Σ.
(6) $\mathbb{M}_B(L \cap (b\Gamma)^*)b\Gamma^A\Omega_A$ *is expressible in* LocTL_Σ.
(7) $\mathbb{M}b(L \cap \Gamma^A)\Omega_A$ *is expressible in* LocTL_Σ.
(8) $\mathbb{M}b\Gamma^A(L \cap \Omega_A)$ *is expressible in* LocTL_Σ.
(9) $L \cap \mathbb{M}b\Gamma^A\Omega_A$ *is expressible in* LocTL_Σ.

In all cases, we use the induction hypothesis to derive a formula for the factor over a smaller alphabet and then *relativize* this formula to the full class. As a representative example, we sketch the proof of Proposition 7(1,2,3).

Proof of Proposition 7(1). The language $L \cap \mathbb{R}_B$ is recognized by $h\restriction_{\mathbb{M}_B}$ (Section 2). By the induction hypothesis, it is expressible in LocTL_B: $L \cap \mathbb{R}_B = \mathcal{L}_B(\alpha) = \{t \in \mathbb{R}_B \mid t \models \alpha\}$ for some $\alpha \in \mathrm{LocTL}_B$. Let $\widetilde{\alpha} \in \mathrm{LocTL}_\Sigma$ be the formula given by Lemma 8 below. We have $(L \cap \mathbb{R}_B) \cdot (\min \subseteq \{b\}) = \mathcal{L}_\Sigma(\widetilde{\alpha})$. □

Lemma 8. *Let* $\alpha \in \mathrm{LocTL}_B$. *There exists a formula* $\widetilde{\alpha} \in \mathrm{LocTL}_\Sigma$ *such that for all* $t = t_1t_2$ *with* $t_1 \in \mathbb{R}_B$ *and* $\min(t_2) \subseteq \{b\}$, $t_1 \models \alpha$ *iff* $t \models \widetilde{\alpha}$.

Proof. We show simultaneously a similar result for internal formulae: for all $\varphi \in \mathrm{LocTL}_B^{\mathrm{i}}$, there exists a formula $\widetilde{\varphi} \in \mathrm{LocTL}_\Sigma^{\mathrm{i}}$ such that for all $t = t_1t_2$ with $t_1 \in \mathbb{R}_B$ and $\min(t_2) \subseteq \{b\}$, and for all $x \in t_1$, $t_1, x \models \varphi$ iff $t, x \models \widetilde{\varphi}$. We use structural induction on α and φ. We have $\widetilde{\alpha \vee \beta} = \widetilde{\alpha} \vee \widetilde{\beta}$, $\widetilde{\neg\alpha} = \neg\widetilde{\alpha}$, $\widetilde{\mathsf{EM}\,\varphi} = \mathsf{EM}(\neg b \wedge \widetilde{\varphi})$, $\widetilde{\varphi \vee \psi} = \widetilde{\varphi} \vee \widetilde{\psi}$, $\widetilde{\neg\varphi} = \neg\widetilde{\varphi}$, $\widetilde{a} = a$, $\widetilde{\mathsf{EX}\,\varphi} = \mathsf{EX}_b\,\widetilde{\varphi}$, $\widetilde{\mathsf{EY}\,\varphi} = \mathsf{EY}\,\widetilde{\varphi}$, $\widetilde{\varphi\,\mathcal{U}_C\,\psi} = \widetilde{\varphi}\,\mathcal{U}_{C\cup\{b\}}\,\widetilde{\psi}$, and $\widetilde{\varphi\,\mathcal{S}_C\,\psi} = \widetilde{\varphi}\,\mathcal{S}_C\,\widetilde{\psi}$. □

To prove Proposition 7(2), we use a reduction to the word case. Let $\mathrm{LTL}_T(\mathsf{XU})$ denote LTL over T-labelled sequences with the strong until modality XU—for $t = t_1t_2\cdots \in T^\infty$, $t \models f\ \mathsf{XU}\ g$ if there exists $j > 1$ such that $t_jt_{j+1}\cdots \models g$ and $t_kt_{k+1}\cdots \models f$ for all $1 < k < j$. From the theory of LTL over sequences, we know that every aperiodic word language $K \in T^\infty$ is expressible in $\mathrm{LTL}_T(\mathsf{XU})$.

We fix $T = h(b\Gamma)$ and define $\sigma : (b\Gamma)^\infty \to T^\infty$ by $\sigma(x) = h(bx_1)h(bx_2)\cdots$ if $x = bx_1bx_2\cdots$ with $x_i \in \Gamma$ for $i \geq 1$. The map σ is well-defined since each trace $x \in (b\Gamma)^\infty$ has a unique factorization $x = bx_1bx_2\cdots$ with $x_i \in \Gamma$ for $i \geq 1$.

We begin with the following result.

Lemma 9. *Let* $L \subseteq \mathbb{R}$ *be recognized by* h. *Then,* $L \cap (b\Gamma)^\infty = \sigma^{-1}(K)$ *for some* K *expressible in* $\mathrm{LTL}_T(\mathsf{XU})$.

Proof. Let $K = \sigma(L \cap (b\Gamma)^\infty)$. We first show that $L \cap (b\Gamma)^\infty = \sigma^{-1}(K)$. The inclusion $\subseteq$ is clear. Conversely, let $x = bx_1bx_2\cdots \in L \cap (b\Gamma)^\infty$ and $y \in \sigma^{-1}(\sigma(x))$. Then, $y = by_1by_2\cdots \in (b\Gamma)^\infty$ with $h(bx_i) = h(by_i)$ for all i. Therefore, $y \sim_h x$ and $x \in L$ implies $y \in L$.

Next, we show that K is recognized by the evaluation morphism $e : T^* \to S$ defined by $e(u) = u$ for all $u \in T$.

Let $s, t \in T^\infty$ with $s \sim_e t$ and $s \in K$. Write $s = s_1s_2\cdots \in T^\infty$ and $t = t_1t_2\cdots \in T^\infty$. Since $T = h(b\Gamma)$ we find $x = bx_1bx_2\cdots \in L$ with $x_i \in \Gamma$ and

$h(bx_i) = s_i$ for all $i > 1$; and $y = by_1by_2\cdots$ with $y_i \in \Gamma$ and $h(by_i) = t_i$ for all $i > 1$. From $s \sim_e t$ we deduce that $x \sim_h y$. Since $x \in L$, $y \in L$ and $t = \sigma(y) \in K$.

Since S is aperiodic, $K \subseteq T^\infty$ is an aperiodic language over T and is thus expressible in $\mathrm{LTL}_T(\mathsf{XU})$—this is the only place where we use S aperiodic. □

The next step is to show that if $K \subseteq T^\infty$ is expressible in $\mathrm{LTL}_T(\mathsf{XU})$ then $\sigma^{-1}(K)$ is expressible in LocTL_Σ.

Lemma 10. *Let $f \in \mathrm{LTL}_T(\mathsf{XU})$. There exists $\widetilde{f} \in \mathrm{LocTL}^{\mathrm{i}}_\Sigma$ such that for all $t = t_1t'$ with $t_1 \in \mathbb{M}$ and $t' \in (b\Gamma)^\infty \setminus \{1\}$, we have $\sigma(t') \models f$ iff $t, \min(t') \models \widetilde{f}$.*

Proof Sketch. We use a structural induction on f. Clearly, $\widetilde{\neg f} = \neg\widetilde{f}$ and $\widetilde{f \vee g} = \widetilde{f} \vee \widetilde{g}$. We claim that $\widetilde{f \,\mathsf{XU}\, g} = \mathsf{EX}\big((\neg b \vee \widetilde{f})\,\mathcal{U}\,(b \wedge \widetilde{g})\big)$.

Let $t' = t_2t_3\cdots$ with $t_i \in b\Gamma$ for all $i > 1$ and let $x = \min(t')$.

Assume that $\sigma(t') \models f \,\mathsf{XU}\, g$ and let $j > 2$ be such that $\sigma(t_jt_{j+1}\cdots) \models g$ and $\sigma(t_kt_{k+1}\cdots) \models f$ for $2 < k < j$. Let $x' \in t$ with $x \lessdot x'$ and $z = \min(t_j)$. We have $x' \le z$, $\lambda(z) = b$ and $t, z \models \widetilde{g}$, by induction. Suppose $y \in t$ with $x' \le y < z$. Either $\lambda(y) \neq b$ and $t, y \models \neg b$, or there exists $2 < k < j$ with $y = \min(t_k)$ and, by induction, $t, y \models \widetilde{f}$. Therefore, $t, x' \models (\neg b \vee \widetilde{f})\,\mathcal{U}\,(b \wedge \widetilde{g})$ and $t, x \models \mathsf{EX}\big((\neg b \vee \widetilde{f})\,\mathcal{U}\,(b \wedge \widetilde{g})\big)$.

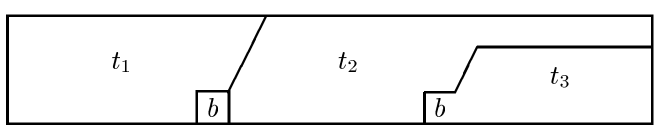
Conversely, assume that $t, x \models \mathsf{EX}((\neg b \vee \widetilde{f})\,\mathcal{U}\,(b \wedge \widetilde{g}))$. Let $x', z \in t$ with $x \lessdot x' \le z$, $t, z \models b \wedge \widetilde{g}$ and $t, y \models \neg b \vee \widetilde{f}$ for all $x' \le y < z$. Since $\lambda(z) = b$, there exists $j > 2$ with $z = \min(t_j)$ and by induction, we get $\sigma(t_jt_{j+1}\cdots) \models g$. Now, let $2 < k < j$ and $y = \min(t_k)$. We have $x' \le y < z$ and since $\lambda(y) = b$ we get $t, y \models \widetilde{f}$ and $\sigma(t_kt_{k+1}\cdots) \models f$ by induction. Therefore, $\sigma(t') \models f \,\mathsf{XU}\, g$.

It remains to deal with a formula of the form $f = s \in T$. For all $r \in S$, the trace language $h_B^{-1}(r) \subseteq \mathbb{M}_B$ is aperiodic and, by induction on $|\Sigma|$, $h_B^{-1}(r) = \mathcal{L}_B(\alpha_r)$ for some formula $\alpha_r \in \mathrm{LocTL}_B$. Let $\widetilde{\alpha_r} \in \mathrm{LocTL}^{\mathrm{i}}_\Sigma$ be the formula obtained using Lemma 11. We claim that $\widetilde{s} = \bigvee_{h(b)\cdot r = s} \widetilde{\alpha_r}$.

We write $t' = bt_2t_3$ with $t_2 \in \Gamma$ and $t_3 \in (b\Gamma)^\infty$. Let $r = h(t_2)$. Then, $t_2 \models \alpha_r$ and by Lemma 11 we obtain $t, \min(t') \models \widetilde{\alpha_r}$. Now, if $\sigma(t') \models s$ then $s = h(bt_2) = h(b)r$ and we get $t, \min(t') \models \widetilde{s}$.

Conversely, assume that $t, \min(t') \models \widetilde{\alpha_r}$ for some r with $s = h(b)r$. By Lemma 11 we get $t_2 \models \alpha_r$ and $h(t_2) = r$. Hence, $h(bt_2) = s$ and $\sigma(t') \models s$. □

Lemma 11. *For all $\alpha \in \mathrm{LocTL}_B$, there is a formula $\widetilde{\alpha} \in \mathrm{LocTL}^{\mathrm{i}}_\Sigma$ such that for all $t = t_1bt_2t_3 \in \mathbb{R}$ with $t_1 \in \mathbb{M}$, $t_2 \in \mathbb{R}_B$, $\min(t_2) \subseteq D(b)$ and $\min(t_3) \subseteq \{b\}$ we have $t_2 \models \alpha$ iff $t, \min(bt_2t_3) \models \widetilde{\alpha}$.*



Proof. We simultaneously establish an equivalent result for internal formulae: for all $\varphi \in \mathrm{LocTL}^{\mathrm{i}}_B$, there exists a formula $\widetilde{\varphi} \in \mathrm{LocTL}^{\mathrm{i}}_\Sigma$ such that for all $t =$

$t_1bt_2t_3 \in \mathbb{R}$ with $t_1 \in \mathbb{M}$, $t_2 \in \mathbb{R}_B$, $\min(t_2) \subseteq D(b)$ and $\min(t_3) \subseteq \{b\}$ and for all $x \in t_2$, we have $t_2, x \models \varphi$ iff $t, x \models \widetilde{\varphi}$.

We proceed by structural induction on α and φ. As always, we have $\widetilde{\neg\alpha} = \neg\widetilde{\alpha}$, $\widetilde{\alpha \vee \beta} = \widetilde{\alpha} \vee \widetilde{\beta}$, $\widetilde{\neg\varphi} = \neg\widetilde{\varphi}$, and $\widetilde{\varphi \vee \psi} = \widetilde{\varphi} \vee \widetilde{\psi}$. Now, $\widetilde{\mathsf{EM}\,\varphi} = \mathsf{EX}(\neg b \wedge \widetilde{\varphi})$, $\widetilde{a} = a$, $\widetilde{\mathsf{EX}\,\varphi} = \mathsf{EX}_b\,\widetilde{\varphi}$, $\widetilde{\mathsf{EY}\,\varphi} = \mathsf{EY}_b(\neg b \wedge \widetilde{\varphi})$,

$$\begin{aligned} \widetilde{\varphi\,\mathcal{U}_C\,\psi} &= \bigvee_{E \subseteq \Sigma} \widetilde{\varphi}\,\mathcal{U}_{(C\cap E)\cup\{b\}}\,(\widetilde{\psi} \wedge E\,\mathcal{S}\,b) \\ \widetilde{\varphi\,\mathcal{S}_C\,\psi} &= \bigvee_{E \subseteq \Sigma} (E\,\mathcal{S}\,b) \wedge (\widetilde{\varphi}\,\mathcal{S}_{(C\cap E)\cup\{b\}}\,(\widetilde{\psi} \wedge \neg b)). \end{aligned}$$

We prove the formula for $\mathcal{U}_C$. Let $x \in t_2$ with $t_2, x \models \varphi\,\mathcal{U}_C\,\psi$. Let $z \in t_2$ be such that $x \le z$, $t_2, z \models \psi$, $t_2, y \models \varphi$ for all $x \le y < z$, and for all $y \in t_2$, $y \le z$ and $\lambda(y) \in C$ implies $y \le x$. By induction, $t, z \models \widetilde{\psi}$ and $t, y \models \widetilde{\varphi}$ for all $x \le y < z$.

Let $E = \mathrm{alph}(\downarrow_{t_2} z) = \mathrm{alph}(\{y \in t_2 \mid y \le z\})$. Clearly, $t, z \models E\,\mathcal{S}\,b$. Let $y \in t$ with $y \le z$. If $\lambda(y) = b$, then $y \le \min(bt_2t_3) \le x$. If $\lambda(y) \in C \cap E$, then there exists $y' \in t_2$ such that $\lambda(y') = \lambda(y) \in C$ and $y \le y'$. From $y' \le z$, we deduce $y' \le x$ and then $y \le y' \le x$.

Conversely, let $x \in t_2$ and $E \subseteq \Sigma$ with $t, x \models \widetilde{\varphi}\,\mathcal{U}_{(C\cap E)\cup\{b\}}\,(\widetilde{\psi} \wedge E\,\mathcal{S}\,b)$. Let $z \in t$ with $x \le z$, $t, z \models \widetilde{\psi} \wedge E\,\mathcal{S}\,b$, $t, y \models \widetilde{\varphi}$ for all $x \le y < z$, and for all $y \le z$, $\lambda(y) \in (C \cap E) \cup \{b\}$ implies $y \le x$. If $z \in t_3$ then $y = \min(t_3)$ satisfies $\lambda(y) = b$, $y \le z$ and $y \not\le x$, a contradiction. Therefore, $z \in t_2$ and by induction, we get $t_2, z \models \psi$ and $t_2, y \models \varphi$ for all $x \le y < z$. Now, let $y \in t_2$ with $y \le z$ and $\lambda(y) \in C$. Since $t, z \models E\,\mathcal{S}\,b$ we deduce that $\mathrm{alph}(\downarrow_{t_2} z) \subseteq E$. Therefore, $\lambda(y) \in C \cap E$ and we obtain $y \le x$. □

Proof of Proposition 7(2). By Lemma 9, $L \cap (b\Gamma)^\infty = \sigma^{-1}(\mathcal{L}_T(f))$ for some $f \in \mathrm{LTL}_T(\mathsf{XU})$. Let $\widetilde{f} \in \mathrm{LocTL}^{\mathrm{i}}_\Sigma$ be given by Lemma 10 and define the formula $\gamma = \mathsf{EM}(b \wedge \widetilde{f}) \vee \mathsf{EM}(\neg b \wedge \mathsf{F}_b\,\mathsf{EX}(b \wedge \widetilde{f}))$. We claim that

$$\mathbb{M}_B \cdot (L \cap (b\Gamma)^\infty \setminus \{1\}) = \mathbb{M}_B \cdot (b\Gamma)^\infty \cap \mathcal{L}(\gamma).$$

Indeed, let $t = t_1t'$ with $t_1 \in \mathbb{M}_B$, $t' \in L \cap (b\Gamma)^\infty \setminus \{1\}$ and let $x = \min(t')$. We have $\sigma(t') \models f$ and by Lemma 10 we get $t, x \models \widetilde{f}$. Now, if $x \in \min(t)$ we have $t \models \mathsf{EM}(b \wedge \widetilde{f})$ and otherwise we have $t \models \mathsf{EM}(\neg b \wedge \mathsf{F}_b\,\mathsf{EX}(b \wedge \widetilde{f}))$.

Conversely, let $t = t_1t'$ with $t_1 \in \mathbb{M}_B$, $t' \in (b\Gamma)^\infty$ and assume that $t \models \gamma$. Necessarily, $b \in \mathrm{alph}(t)$ and therefore $t' \ne 1$. From $t \models \gamma$ we deduce that $t, x \models b \wedge \widetilde{f}$ with $x = \min(t')$. By Lemma 10 we get $\sigma(t') \models f$. Therefore, $t' \in \sigma^{-1}(\mathcal{L}_T(f)) \subseteq L$ and t belongs to the left-hand side.

Since $\mathbb{M}_B \cdot (b\Gamma)^\infty$ and $\mathbb{M}_B$ are both expressible in LocTL_Σ, we deduce that $\mathbb{M}_B \cdot (L \cap (b\Gamma)^\infty)$ is expressible in LocTL_Σ. □

Proof of Proposition 7(3). From Lemma 4, $L \cap \mathbb{M}_B(b\Gamma)^\infty$ is a finite union of languages of the form $(L_1 \cap \mathbb{M}_B) \cdot (L_2 \cap (b\Gamma)^\infty)$ where the trace languages $L_i \subseteq \mathbb{R}$

are recognized by h. Now, since the product $\mathbb{M}_B(b\Gamma)^\infty$ is unambiguous, we have

$$(L_1 \cap \mathbb{M}_B) \cdot (L_2 \cap (b\Gamma)^\infty) = (L_1 \cap \mathbb{M}_B) \cdot (b\Gamma)^\infty \cap \mathbb{M}_B \cdot (L_2 \cap (b\Gamma)^\infty)$$
$$(L_1 \cap \mathbb{M}_B) \cdot (b\Gamma)^\infty = (\mathbb{M}_B \cdot (b\Gamma)^\infty) \cap ((L_1 \cap \mathbb{R}_B) \cdot (\min \subseteq \{b\})).$$

We conclude using Lemma 3 and Proposition 7(1,2). □

We can prove Proposition 7(1-3) using no EY and only sinces of the form $A\,\mathcal{S}_C\,a$ ("alphabetic" sinces). If $L \subseteq \mathbb{M}$ then $L = L \cap \mathbb{M}_B(b\Gamma)^*$, so we only need to consider the first case in the decomposition. From this, we have the following.

Theorem 12. *Let $L \subseteq \mathbb{M}$ be a trace language recognized by h. Then, L is expressible in* LocTL_Σ *without using* EY *and with alphabetic sinces only.*

6 Satisfiability

To decide satisfiability, we work with alternating automata over sequentializations of traces. The states of the alternating automaton consist of subformulas of the main formula together with bookkeeping information about the letters traversed while satisfying until and since requirements. The basic technique we use was introduced in [1] and refined in [4]. The additional complication for our logic is that we have to deal with past requirements, both explicitly in the operators EY and $\mathcal{S}_C$, and implicitly, because of the alphabetic filters in $\mathcal{U}_C$ and $\mathcal{S}_C$. We thus need two-way alternating automata. Nevertheless, checking emptiness for this class remains within PSPACE [10] for each fixed dependence alphabet (Σ, D). Due to space constraints, we omit all details of the construction.

7 Open problems

As we have seen in Section 3, the introduction of past modalities is unavoidable when constructing an expressively complete local temporal logic over traces. A natural question is whether unfiltered versions of $\mathcal{U}_C$ and $\mathcal{S}_C$ are sufficient for expressive completeness. We also do not know whether the temporal logic based only on EX and $\mathcal{U}_C$ is expressively complete. Another alternative is to keep the unfiltered until and to use a concurrency modality asserting the existence of a concurrent vertex satisfying some formula. Also, we have seen in Theorem 12 that no EY are required for finite traces and that alphabetic sinces are sufficient. We do not know whether the same restriction applies to infinite traces.

References

1. R. Alur, D. Peled, and W. Penczek. Model-checking of causality properties. In *Proceedings of LICS'95*, pages 90–100, 1995.
2. C. Courcoubetis, M.Y. Vardi, P. Wolper, and M. Yannakakis. Memory efficient algorithms for the verification of temporal properties. *formal Methods in System Design*, 1:275–288, 1992.

3. V. Diekert and P. Gastin. LTL is expressively complete for Mazurkiewicz traces. In *Proceedings of ICALP 2000*, number 1853 in LNCS, pages 211–222. Springer Verlag, 2000.
4. V. Diekert and P. Gastin. Local temporal logic is expressively complete for cograph dependence alphabets. In *Proceedings of LPAR'01*, number 2250 in LNAI, pages 55–69. Springer Verlag, 2001.
5. V. Diekert and P. Gastin. LTL is expressively complete for Mazurkiewicz traces. *Journal of Computer and System Sciences*, 2002. To appear.
6. V. Diekert and G. Rozenberg, editors. *The Book of Traces.* World Scientific, Singapore, 1995.
7. W. Ebinger and A. Muscholl. Logical definability on infinite traces. *Theoretical Computer Science*, 154:67–84, 1996.
8. W. Ebinger. *Charakterisierung von Sprachklassen unendlicher Spuren durch Logiken.* Dissertation, Institut für Informatik, Universität Stuttgart, 1994.
9. D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *Proceedings of PoPL'80*, pages 163–173, Las Vegas, Nev., 1980.
10. T. Jiang and B. Ravikumar. A note on the space complexity of some decision problems for finite automata. *Information Processing Letters*, 40:25–31, 1991.
11. J.A.W. Kamp. *Tense Logic and the Theory of Linear Order.* PhD thesis, University of California, Los Angeles, California, 1968.
12. A. Mazurkiewicz. Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University, Aarhus, 1977.
13. P. Niebert. A ν-calculus with local views for sequential agents. In *Proceedings of MFCS'95*, number 969 in LNCS, pages 563–573. Springer Verlag, 1995.
14. D. Perrin and J.-E. Pin. Infinite words. Technical report, LITP, Avril 1997.
15. A. Pnueli. The temporal logics of programs. In *Proceedings of the 18th IEEE FOCS, 1977*, pages 46–57, 1977.
16. R. Ramanujam. Locally linear time temporal logic. In *Proceedings of LICS'96*, pages 118–128, 1996.
17. P.S. Thiagarajan and I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. In *Proceedings of LICS'97*, pages 183–194, 1997.
18. P.S. Thiagarajan. A trace based extension of linear time temporal logic. In *Proceedings of LICS'94*, pages 438–447, 1994.
19. M.Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Proceedings of LICS'86*, pages 322–331, 1986.
20. M.Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Logics for Concurrency: Structure versus Automata*, number 1043 in LNCS, pages 238–266. Springer Verlag, 1996.
21. I. Walukiewicz. Difficult configurations – on the complexity of LTrL. In *Proceedings of ICALP'98*, number 1443 in LNCS, pages 140–151. Springer Verlag, 1998.
22. I. Walukiewicz. Private communication, 2001.
23. I. Walukiewicz. Local logics for traces. *Journal of Automata, Languages and Combinatorics*, 2002. To appear.
24. Th. Wilke. Classifying discrete temporal properties. Habilitationsschrift (post-doctoral thesis), April 1998.
25. Th. Wilke. Classifying discrete temporal properties. In *Proceedings of STACS'99*, number 1563 in LNCS, pages 32–46. Springer Verlag, 1999.
26. W. Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O. — Informatique Théorique et Applications*, 21:99–135, 1987.