Available online at www.sciencedirect.com

ScienceDirect

Information and Computation 204 (2006) 1597–1619

Information and Computation

www.elsevier.com/locate/ic

Pure future local temporal logics are expressively complete for Mazurkiewicz traces ☆

Volker Diekert [a], Paul Gastin [b,*]

[a] *Institut für Formale Methoden der Informatik (FMI), Universität Stuttgart, Universitätsstr. 38, D-70569 Stuttgart, Germany*

[b] *LSV, ENS de Cachan & CNRS, 61 Av. du Président Wilson, F-94235 Cachan Cedex, France*

Received 24 June 2005
Available online 25 September 2006

Abstract

The paper settles a long standing problem for Mazurkiewicz traces: the *pure future local* temporal logic defined with the basic modalities *exists-next* and *until* is expressively complete. This means every first-order definable language of Mazurkiewicz traces can be defined in a pure future local temporal logic. The analogous result with a global interpretation has been known, but the treatment of a local interpretation turned out to be much more involved. Local logics are interesting because both the satisfiability problem and the model checking problem are solvable in PSPACE for these logics whereas they are non-elementary for global logics. Both, the (previously known) global and the (new) local results generalize Kamp's Theorem for words, because for sequences local and global viewpoints coincide.

Keywords: Temporal logics; Mazurkiewicz traces; Concurrency

☆ Work partly done while the second author stayed in Stuttgart within the excellence programme MERCATOR of the German Research Foundation DFG. Partial support of ACI Sécurité Informatique 2003-22 (VERSYDIS) is gratefully acknowledged.

* Corresponding author.

E-mail addresses: diekert@fmi.uni-stuttgart.de (V. Diekert), Paul.Gastin@lsv.ens-cachan.fr (P. Gastin).

0890-5401/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2006.07.002

1. Introduction

In the middle of the 1970's Mazurkiewicz proposed *trace theory* as an algebraic framework for studying concurrent processes [22]. Based on the early work of Keller [19] he described the behaviour of a concurrent process not by a string, but more accurately by some labelled partial order which is called a *trace*. The partial order relation of a trace is defined via a static dependence relation so that the set of traces forms a *free partially commutative monoid*. There is a natural extension to infinite objects which lead to a notion of *real trace*. For an overview on trace theory we refer to *The Book of Traces* [9].

One advantage of trace theory is that formal specifications of concurrent systems by temporal logic formulae have a direct (either global or local) interpretation for Mazurkiewicz traces. It is therefore no surprise that temporal logics for traces have received quite an attention, see [26,32,10,2,25,24,29].

For a global interpretation it was shown by Thiagarajan and Walukiewicz [33] that the global temporal logic with future modalities and with past constants is expressively complete with respect to the first order theory. In [4] we were able to remove the past constants using an algebraic proof. However, the satisfiability problem for these global logics is non-elementary [35]. The main reason for this high complexity is that the interpretation of a formula is defined with respect to a global configuration, i.e., a finite prefix of the trace (downward closed subset of the partial order which defines the trace)—and the prefix structure of traces is much more complex than in the case of linear orders (words).

In contrast to a global formula, a local logic formula is evaluated at a local event of the system, i.e., at some vertex of the trace. There can be exponentially many different configurations in a finite trace, but the number of vertices is just the length of the trace. This makes local model checking much easier. In fact, if the underlying alphabet is fixed, all local temporal logics over traces where the modalities are definable in monadic second order logic are decidable in Pspace [14] (both the satisfiability problem and the model checking problem are decidable in Pspace). This is optimal since the Pspace-hardness occurs already for words (over a two letter alphabet).

The better complexity makes local temporal logics more attractive than global ones; and several attempts were made to prove expressive completeness with respect to first-order logic. In [6] expressive completeness for the basic pure future local temporal logic is established, if the underlying dependence alphabet is a cograph, i.e., if the modelled system can be obtained using series and parallel compositions. Moreover, one can hope to go beyond cographs, only if each trace is equipped with some bottom element or if we allow past modalities. This second approach is used in [15,16] to obtain expressive completeness for all dependence alphabet. In [15], the full power of *exists-previous* and *since* modalities equipped with filters is used. The result is improved in [16] where only past constants are necessary. Another temporal logic based on more involved modalities (including both past and future modalities) was shown to be expressively complete and decidable in Pspace [1]. However, the most basic question remained open: whether expressive completeness holds for a pure future local temporal logic.

The present paper gives a positive answer to this question. It is well-known that first-order definable trace languages are aperiodic. Here, we give a self-contained proof that every aperiodic trace language is definable in a pure future local temporal logic based upon *exists-next* and *until*, only. The well-known corresponding result for words is not used in the proof, formally it becomes a corollary. We also show that a pure future process-based logic in the spirit of the logic TrPTL introduced by Thiagarajan in [32] is expressively complete.

Our proof is inspired by Wilke's proof for the corresponding result on finite words [37]. It is actually a generalization since it deals with both finite and infinite traces, in particular it includes infinite words. It also simplifies Wilke's technique thanks to some non-standard construction on finite monoids, which allows to use as a main induction parameter the size of the monoid and therefore avoids the deviation via transformation monoids.

An extended abstract of a preliminary version of this paper appeared in [7].

2. Preliminaries

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 dependent on A is denoted by $D(A) = \{b \in \Sigma \mid (a,b) \in D \text{ for some } a \in A\}$.

A *Mazurkiewicz trace* is an equivalence class of a labelled partial order $t = [V, \leqslant, \lambda]$ where V is a set of vertices labelled by $\lambda : V \to \Sigma$ and $\leqslant$ is a partial order over V satisfying the following three conditions: for all $x \in V$, the downward closed set $\downarrow x = \{y \in V \mid y \leqslant x\}$ is finite, for all $x, y \in V$, $(\lambda(x), \lambda(y)) \in D$ implies $x \leqslant y$ or $y \leqslant x$, and if x is an immediate predecessor of y, then $(\lambda(x), \lambda(y)) \in D$. In the following $\lessdot$ denotes the immediate predecessor relation in V, i.e., $\lessdot = < \setminus <^2$ and the last condition says that $x \lessdot y$ implies $(\lambda(x), \lambda(y)) \in D$. For $x \in V$, we also define the upper set $\uparrow x = \{y \in V \mid x \leqslant y\}$ and the strict upper set $\Uparrow x = \{y \in V \mid x < y\}$.

Since the alphabet is finite, we have an equivalent definition of a Mazurkiewicz trace t as follows: We start with a finite or infinite word $a_1 a_2 \cdots$ where all a_i are letters in Σ. Each i is viewed as a node of a labelled graph and the node i has label $\lambda(i) = a_i$. We draw an arc from a_i to a_j if and only if both, $i < j$ and $(a_i, a_j) \in D$. We obtain a directed acyclic graph and $t = [V, \leqslant, \lambda]$ is defined as the induced labelled partial order. In particular, every trace t has a representation by some word $a_1 a_2 \cdots \in \Sigma^\infty$.

A trace t is called *finite* (*infinite* resp.) if V is finite (infinite resp.), and we denote by $\mathbb{M}(\Sigma, D)$ (or simply $\mathbb{M}$) the set of finite traces. By $\mathbb{R}(\Sigma, D)$ (or simply $\mathbb{R}$), we denote the set of finite or infinite traces (also called *real traces*). Let $\text{alph}(t) = \lambda(V)$ be the *alphabet* of t and $\text{alphinf}(t) = \{a \in \Sigma \mid \lambda^{-1}(a) \text{ is infinite}\}$ be the *alphabet at infinity* of t. For $A \subseteq \Sigma$, we let $\mathbb{R}_A = \{t \in \mathbb{R} \mid \text{alph}(t) \subseteq A\}$ and $\mathbb{M}_A = \{t \in \mathbb{M} \mid \text{alph}(t) \subseteq A\}$.

Let $t_1 = [V_1, \leqslant_1, \lambda_1]$ and $t_2 = [V_2, \leqslant_2, \lambda_2]$ be a pair of traces such that $\text{alphinf}(t_1) \times \text{alph}(t_2) \subseteq I$. Then we define the concatenation of t_1 and t_2 to be $t_1 \cdot t_2 = [V, \leqslant, \lambda]$ where $V = V_1 \cup V_2$ (assuming w.l.o.g. that $V_1 \cap V_2 = \emptyset$), $\lambda = \lambda_1 \cup \lambda_2$, and $\leqslant$ is the transitive closure of the relation $\leqslant_1 \cup \leqslant_2 \cup (V_1 \times V_2 \cap \lambda^{-1}(D))$. The set $\mathbb{M}$ of finite traces is then a monoid with the empty trace $1 = (\emptyset, \emptyset, \emptyset)$ as unit. If we can write $t = rs$, then r is a prefix and s is a suffix of t. Note that a factorization of a real trace $t \in \mathbb{R}$ may yield an infinite prefix and/or suffix. Consider e.g., $t = (ab)^\omega = (a^\omega)(b^\omega)$ with $(a,b) \in I$. The concatenation of two trace languages $K, L \subseteq \mathbb{R}$ is $K \cdot L = \{r \cdot s \mid r \in K, s \in L \text{ and } \text{alphinf}(r) \times \text{alph}(s) \subseteq I\}$. We also use finite or infinite (ordered) products $t = \prod_{i \in J} t_i$, where $(t_i)_{i \in J}$ is a sequence of real traces with $J \subseteq \mathbb{N}$ and $t_i \in \mathbb{R}$ such that $\text{alphinf}(t_i) \times \text{alph}(t_j) \subseteq I$ for all $i < j$.

We denote by $\min(t)$ the set of minimal vertices of t. We let $\mathbb{R}^1 = \{t \in \mathbb{R} \mid |\min(t)| = 1\}$ be the set of traces with exactly one minimal vertex. To simplify the notation, we also use $\min(t)$ for the set $\lambda(\min(t))$ of labels of the minimal vertices of t.

The syntax of first-order logic $\mathrm{FO}_\Sigma(<)$ is defined as follows:

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

where $a \in \Sigma$, and $x, y \in \mathbb{V}$ are first order variables. We use the standard semantics. Given a trace $t = [V, \leqslant, \lambda]$ and a valuation $\nu : \mathbb{V} \to V$, $t \models_\nu \varphi$ denotes that t satisfies φ under ν. 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 formulae as usual. The meaning of a closed formula (*sentence*) φ is independent of the valuation ν, hence the subscript ν can be suppressed. We say that a real trace language $L \subseteq \mathbb{R}$ is expressible in $\mathrm{FO}_\Sigma(<)$, if there exists a sentence $\varphi \in \mathrm{FO}_\Sigma(<)$ such that $L = \{t \in \mathbb{R} \mid t \models \varphi\}$.

3. Local temporal logic

We want to compare the expressive power of local temporal logics with the first order logic $\mathrm{FO}_\Sigma(<)$. Our main focus is on the local temporal logic based upon the two classical modalities *exists-next* and *until*. The syntax of the local temporal logic $\mathrm{LocTL}_\Sigma[\mathsf{EX}, \mathsf{U}]$ is given by

$$\varphi ::= \top \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi,$$

where a ranges over Σ and $\top$ denotes *true*.

Let $t = [V, \leqslant, \lambda] \in \mathbb{R}$ be a real trace and $x \in V$ be a vertex. (We write henceforth simply $x \in t$ instead of $x \in V$.) We define the semantics such that every temporal formula is equivalent to some first-order formula with one free variable and using at most three distinct variables.

$$\begin{array}{ll}
t,x \models \top & \\
t,x \models a & \text{if } \lambda(x) = a \\
t,x \models \neg\varphi & \text{if } t,x \not\models \varphi \\
t,x \models \varphi \vee \psi & \text{if } t,x \models \varphi \text{ or } t,x \models \psi \\
t,x \models \mathsf{EX}\,\varphi & \text{if } \exists y(x{<}y \text{ and } t,y \models \varphi) \\
t,x \models \varphi\,\mathsf{U}\,\psi & \text{if } \exists z(x \leqslant z \text{ and } t,z \models \psi \text{ and } \forall y(x \leqslant y < z) \Rightarrow t,y \models \varphi).
\end{array}$$

For $t \in \mathbb{R}^1$, i.e., if t has a unique minimal vertex, we simply write $t \models \varphi$ instead of $t, \min(t) \models \varphi$.

We define some abbreviations. We write $\bot$ for *false*. The formula $\mathsf{F}\,\varphi = \top\,\mathsf{U}\,\varphi$ means that φ holds now or at some position in the future and the formula $\mathsf{G}\,\varphi = \neg\,\mathsf{F}\,\neg\varphi$ means that φ holds at all future positions, including the current one. For $A \subseteq \Sigma$, we also use A as a formula with the definition $A = \bigvee_{a \in A} a$.

The two modalities *exists-next* and *until* can be expressed by a single one, the *strict-until* modality SU, the semantics of which is given by

$$t,x \models \varphi\,\mathsf{SU}\,\psi \text{ if } \exists z(x < z \text{ and } t,z \models \psi \text{ and } \forall y(x < y < z) \Rightarrow t,y \models \varphi).$$

We have $\mathsf{EX}\,\varphi = \bot\,\mathsf{SU}\,\varphi$ and $\varphi\,\mathsf{U}\,\psi = \psi \vee (\varphi \wedge \varphi\,\mathsf{SU}\,\psi)$. Thus, $\mathrm{LocTL}_\Sigma[\mathsf{EX},\mathsf{U}]$ is clearly a fragment of $\mathrm{LocTL}_\Sigma[\mathsf{SU}]$.

We do not know any direct way how to express SU in $\mathrm{LocTL}_\Sigma[\mathsf{EX},\mathsf{U}]$. But it follows from our main result Corollary 26 that the two logics $\mathrm{LocTL}_\Sigma[\mathsf{EX},\mathsf{U}]$ and $\mathrm{LocTL}_\Sigma[\mathsf{SU}]$ have the same expressive power. Note that, if $D = \Sigma \times \Sigma$, i.e., if we are in the classical situation of words, then $\varphi\,\mathsf{SU}\,\psi$ and $\mathsf{EX}(\varphi\,\mathsf{U}\,\psi)$ are equivalent, hence we get easily the equivalence of the two logics for words. But as soon as there are letters a, b, c with $(a,b) \in D$, $(b,c) \in D$, and $(a,c) \in I$ then $\varphi\,\mathsf{SU}\,\psi$ is not equivalent with $\mathsf{EX}(\varphi\,\mathsf{U}\,\psi)$. Consider for instance the trace $t = bacb$. We have $t \models \mathsf{EX}(a\,\mathsf{U}\,b)$ but $t \not\models a\,\mathsf{SU}\,b$.

We need some more notations. For $x \in t$ and $c \in \mathrm{alph}(\Uparrow x)$, we denote by x_c the unique minimal vertex of $\Uparrow x \cap \lambda^{-1}(c)$. Note that $x < x_c$, if x_c exists. We write $x_a \parallel x_b$, if both vertices x_a and x_b exist, but neither $x_a \leqslant x_b$ nor $x_a \geqslant x_b$.

Let us define some more operators that turn out to be crucial to achieve our main result. We will see that all of them can be expressed in $\mathrm{LocTL}_\Sigma[\mathsf{EX},\mathsf{U}]$. Let $a, b \in \Sigma$. The semantics of the operators $(\mathsf{X}_a \leqslant \mathsf{X}_b)$, $(\mathsf{X}_a < \mathsf{X}_b)$, $(\mathsf{X}_a \parallel \mathsf{X}_b)$, X_a and U_a is defined as follows:

$$\begin{array}{ll}
t,x \models (\mathsf{X}_a \leqslant \mathsf{X}_b) & \text{if } x_a, x_b \text{ exist and } x_a \leqslant x_b\\
t,x \models (\mathsf{X}_a < \mathsf{X}_b) & \text{if } x_a, x_b \text{ exist and } x_a < x_b\\
t,x \models (\mathsf{X}_a \parallel \mathsf{X}_b) & \text{if } x_a, x_b \text{ exist and } x_a \parallel x_b\\
t,x \models \mathsf{X}_a\,\varphi & \text{if } x_a \text{ exists and } t, x_a \models \varphi\\
t,x \models \varphi\,\mathsf{U}_a\,\psi & \text{if } \exists z (x \leqslant z \text{ and } \lambda(z) = a \text{ and } t,z \models \psi \text{ and}\\
& \qquad\qquad \forall y (x \leqslant y < z \text{ and } \lambda(y) = a) \Rightarrow t,y \models \varphi).
\end{array}$$

We now introduce the logic $\mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ which plays the central role in the following. Its syntax is given by

$$\varphi ::= \top \mid a \mid (\mathsf{X}_a \leqslant \mathsf{X}_b) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}_a\,\varphi \mid \varphi\,\mathsf{U}_a\,\varphi,$$

where a, b range over Σ. The semantics has been defined above.

Note that $\mathsf{F}\,\varphi$, $(\mathsf{X}_a < \mathsf{X}_b)$, and $(\mathsf{X}_a \parallel \mathsf{X}_b)$ can easily be expressed in the logic $\mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$, so we can freely use them. For instance, $\mathsf{F}\,\varphi = \bigvee_a \top\,\mathsf{U}_a\,\varphi$ and $(\mathsf{X}_a \parallel \mathsf{X}_b) = \mathsf{X}_a \top \wedge \mathsf{X}_b \top \wedge \neg(\mathsf{X}_a \leqslant \mathsf{X}_b) \wedge \neg(\mathsf{X}_b \leqslant \mathsf{X}_a)$.

We show that we can deal also with process-based logics as introduced in [32]. In this framework, we start with a finite set of processes $\mathcal{P} = \{1, \ldots, n\}$ and a mapping $p : \Sigma \to 2^{\mathcal{P}} \setminus \{\emptyset\}$. If $p(a) = \{i\}$ is a singleton then the action a is local to process i. Otherwise, the execution of a requires the synchronization of all processes in $p(a)$. The dependence relation is therefore $D = \{(a,b) \in \Sigma^2 \mid p(a) \cap p(b) \neq \emptyset\}$. In the following, we let $\Sigma_i = \{a \in \Sigma \mid i \in p(a)\}$. The set $\mathcal{C} = \{\Sigma_i \mid i \in \mathcal{P}\}$ is a covering of Σ by cliques of (Σ, D).

Note that every dependence relation D can be obtained this way. We may use for the set $\mathcal{P}$ any covering of (Σ, D) by cliques and let $p(a) = \{C \in \mathcal{P} \mid a \in C\}$.

Thanks to this more concrete view of the dependence alphabet based on processes, we can define temporal modalities that involve locations of actions as in [32]. However (c.f. Remark 1) we focus on pure future variants $\mathsf{X}_i\,\varphi$ meaning that φ holds at the first event of process i which is strictly above the current vertex and $\varphi \mathsf{U}_i \psi$ which means that on the sequence of vertices located on process i and above the current vertex we observe φ until ψ.

More formally, we introduce the logic $\mathrm{LocTL}_\Sigma[\mathsf{X}_i, \mathsf{U}_i]$ based on the modalities X_i and U_i for $i \in \mathcal{P}$ by the syntax

$$\varphi ::= \top \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}_i\,\varphi \mid \varphi\,\mathsf{U}_i\,\varphi,$$

where a, b range over Σ and i ranges over $\mathcal{P}$.

For $x \in t$ and $i \in \mathcal{P}$, we denote by x_i the unique minimal vertex of $\Uparrow x \cap \lambda^{-1}(\Sigma_i)$ if it exists, i.e., when $\Uparrow x \cap \lambda^{-1}(\Sigma_i) \neq \emptyset$. The semantics of the new modalities is given by

$$\begin{array}{ll} t,x \models \mathsf{X}_i\,\varphi & \text{if } x_i \text{ exists and } t,x_i \models \varphi \\ t,x \models \varphi\,\mathsf{U}_i\,\psi & \text{if } \exists z(x \leqslant z \text{ and } \lambda(z) \in \Sigma_i \text{ and } t,z \models \psi \text{ and} \\ & \qquad \forall y(x \leqslant y < z \text{ and } \lambda(y) \in \Sigma_i) \Rightarrow t,y \models \varphi). \end{array}$$

Note that U_i is a usual *sequential* until on the chain of vertices $\uparrow x \cap \lambda^{-1}(\Sigma_i)$.

Remark 1. In [32], the formula $\mathcal{O}_i\varphi$ means that φ holds at the first event of process i that is not in the past of the current vertex. Clearly, this is not a future modality. The until modality introduced in [32] is also not pure future. This motivates our different choice.

Proposition 2. *The expressiveness of the following local temporal logics is increasing* (*or equal*) *in the following order*:

(1) $\mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$,
(2) $\mathrm{LocTL}_\Sigma[\mathsf{X}_i, \mathsf{U}_i]$,
(3) $\mathrm{LocTL}_\Sigma[\mathsf{EX}, \mathsf{U}]$,
(4) $\mathrm{LocTL}_\Sigma[\mathsf{SU}]$.

Proof. (1) $\subseteq$ (2): Fix $a \in \Sigma$ and let $i \in \mathcal{P}$ with $i \in p(a)$. We have $\mathsf{X}_a\,\varphi = \mathsf{X}_i(\neg a\,\mathsf{U}_i\,(a \wedge \varphi))$ and $\varphi\,\mathsf{U}_a\,\psi = (\neg a \vee \varphi)\,\mathsf{U}_i\,(a \wedge \psi)$.

We show now how to express the constants $(\mathsf{X}_a \leqslant \mathsf{X}_b)$, which is more difficult. The idea is that $t,x \models (\mathsf{X}_a \leqslant \mathsf{X}_b)$ if and only if there exists a chain $x_0, \ldots, x_n$ in t with $n \leqslant |\Sigma|$ and $x_a = x_0 < x_1 < \cdots < x_n = x_b$ and $(\lambda(x_i), \lambda(x_{i+1})) \in D$ for $0 \leqslant i < n$.

For this, we define inductively formulae $(\mathsf{X}_a \leqslant_n \mathsf{X}_b)$ by:

$$(\mathsf{X}_a \leqslant_1 \mathsf{X}_b) = \begin{cases} \bot & \text{if}\,(a,b) \in I \\ \mathsf{X}_i((\top\,\mathsf{U}_i\,b) \wedge (\neg b\,\mathsf{U}_i\,a)) & \text{otherwise, where } i \in p(a) \cap p(b) \end{cases}$$

and for $n > 1$, we define $(\mathsf{X}_a \leqslant_n \mathsf{X}_b)$ by

$$\begin{aligned} &(\mathsf{X}_a \leqslant_{n-1} \mathsf{X}_b) \vee \bigvee_{c \in D(a) \setminus \{a,b\}} \Big[\big((\mathsf{X}_a \leqslant_1 \mathsf{X}_c) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)\big) \vee \\ &\quad \Big[\big((\mathsf{X}_c \leqslant_1 \mathsf{X}_a) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)\big) \wedge \\ &\quad \big((\mathsf{X}_c \leqslant_1 \mathsf{X}_a) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)\big)\,\mathsf{U}_c\,\big((\mathsf{X}_a \leqslant_1 \mathsf{X}_c) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)\big)\Big]\Big]. \end{aligned}$$

We claim that $(\mathsf{X}_a \leqslant \mathsf{X}_b) = (\mathsf{X}_a \leqslant_{|\Sigma|} \mathsf{X}_b)$.

We first show by induction on n that $(\mathsf{X}_a \leqslant_n \mathsf{X}_b)$ implies $(\mathsf{X}_a \leqslant \mathsf{X}_b)$. Fix $t \in \mathbb{R}$ and $x \in t$. Assume first that $t,x \models (\mathsf{X}_a \leqslant_1 \mathsf{X}_b)$. Then, $t,x \models \mathsf{X}_i((\top \mathsf{U}_i\, b) \wedge (\neg b \,\mathsf{U}_i\, a))$ for some $i \in p(a) \cap p(b)$. We deduce easily that $t,x \models (\mathsf{X}_a \leqslant \mathsf{X}_b)$. Now, let $n > 1$, $c \in \Sigma$ and assume that $t,x \models (\mathsf{X}_a \leqslant_1 \mathsf{X}_c) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)$. By induction, we get $t,x \models (\mathsf{X}_a \leqslant \mathsf{X}_c) \wedge (\mathsf{X}_c \leqslant \mathsf{X}_b)$ which implies clearly $t,x \models (\mathsf{X}_a \leqslant \mathsf{X}_b)$. Finally, assume that $t,x \models (\mathsf{X}_c \leqslant_1 \mathsf{X}_a) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)$ and $t,x \models \big((\mathsf{X}_c \leqslant_1 \mathsf{X}_a) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)\big)\, \mathsf{U}_c \,\big((\mathsf{X}_a \leqslant_1 \mathsf{X}_c) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)\big)$. Let z be such that $x \leqslant z$, $\lambda(z) = c$, $t,z \models (\mathsf{X}_a \leqslant_1 \mathsf{X}_c) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)$ and $t,y \models (\mathsf{X}_c \leqslant_1 \mathsf{X}_a) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)$ for each $x \leqslant y < z$ with $\lambda(y) = c$. By induction we get $t,z \models (\mathsf{X}_a \leqslant \mathsf{X}_c) \wedge (\mathsf{X}_c \leqslant \mathsf{X}_b)$ which implies $t,z \models (\mathsf{X}_a \leqslant \mathsf{X}_b)$. It remains to show that $x_a = z_a$ and $x_b = z_b$. Let $y_1,\ldots,y_k$ be the c-labelled vertices between x and z with $x = y_0 < y_1 < \cdots < y_k = z$. For $0 \leqslant i < k$ we have $t,y_i \models (\mathsf{X}_c \leqslant_1 \mathsf{X}_a) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)$ and by induction we get $(y_i)_c < (y_i)_a$ and $(y_i)_c < (y_i)_b$ (recall that $c \notin \{a,b\}$). Since we also have $(y_i)_c = y_{i+1}$, we deduce that $(y_i)_a = (y_{i+1})_a$ and $(y_i)_b = (y_{i+1})_b$. Using $x = y_0$ and $z = y_k$ we obtain $x_a = z_a$ and $x_b = z_b$. Therefore, $x_a = z_a < z_b = x_b$ as desired.

Conversely, we show by induction on n that for $t \in \mathbb{R}$ and $x \in t$, if x_a, x_b exist and there exist $x_0,\ldots,x_n$ with $x_a = x_0 < x_1 < \cdots < x_n = x_b$ and $(\lambda(x_i),\lambda(x_{i+1})) \in D$ for $0 \leqslant i < n$ and n is minimal with this property, then $t,x \models (\mathsf{X}_a \leqslant_n \mathsf{X}_b)$.

Consider first the case $n = 1$. Then $(a,b) \in D$ and if $i \in p(a) \cap p(b)$ we obtain easily $t,x \models \mathsf{X}_i((\top \,\mathsf{U}_i\, b) \wedge (\neg b \,\mathsf{U}_i\, a))$.

Assume now $n > 1$. Since n is minimal, we have $c = \lambda(x_1) \in D(a) \setminus \{a,b\}$. Without loss of generality, we may assume that $x_1 = (x_a)_c$. Let $y_1,\ldots,y_k$ be the c-labelled vertices between x and x_1 with $x = y_0 < y_1 < \cdots < y_k = x_1$. If $k = 1$ then $x_1 = x_c$ and $x_a < x_c$ hence we get $t,x \models (\mathsf{X}_a \leqslant_1 \mathsf{X}_c)$. By induction we also get $t,x \models (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)$ (with $x_2,\ldots,x_n$). Therefore, $t,x \models (\mathsf{X}_a \leqslant_n \mathsf{X}_b)$. Assume now $k > 1$. Since $(a,c) \in D$, we must have y_{k-1} and x_a ordered. If $x_a < y_{k-1}$ then $y_k = x_1 = (x_a)_c \leqslant y_{k-1}$, a contradiction. Therefore, $y_{k-1} < x_a$. With $z = y_{k-1}$ we obtain $z_a = x_a$. Since $x_a < x_b$ we also get $z_b = x_b$. For $0 \leqslant i < k$ we have $(y_i)_c = y_{i+1}$ and in particular $z_c = y_k = x_1$. Therefore, $z_a < z_c < z_b$ and we get by induction $t,z \models (\mathsf{X}_a \leqslant_1 \mathsf{X}_c) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)$ (with $x_2,\ldots,x_n$). Finally, let $0 \leqslant i < k-1$. We have $x \leqslant y_i < (y_i)_c = y_{i+1} \leqslant y_{k-1} < x_a$ and we deduce $(y_i)_a = x_a > (y_i)_c$. Since $x_a < x_b$ we also get $(y_i)_b = x_b > (y_i)_c$. By induction we get $t,y_i \models (\mathsf{X}_c \leqslant_1 \mathsf{X}_a) \wedge (\mathsf{X}_c \leqslant_{n-1} \mathsf{X}_b)$ (with $x_2,\ldots,x_n$). Therefore, $t,x \models (\mathsf{X}_a \leqslant_n \mathsf{X}_b)$.

This concludes the proof of our claim since whenever $x < y$ in a trace then we find a path $x = x_0 < x_1 < \cdots < x_n = y$ with $(\lambda(x_i),\lambda(x_{i+1})) \in D$ of length at most $|\Sigma|$. Actually, we have $(\mathsf{X}_a \leqslant \mathsf{X}_b) = (\mathsf{X}_a \leqslant_k \mathsf{X}_b)$ where k is the maximal length of a simple path in the dependence alphabet (Σ,D) for $a \neq b$, and where $k = 1$ for $a = b$.

(1) $\subseteq$ (3): We include this part because in order to prove (2) $\subseteq$ (3) we will use the constants $(\mathsf{X}_a \leqslant \mathsf{X}_b)$ and the modality X_a, hence we show first how to express them in $\mathrm{LocTL}_\Sigma(\mathsf{EX},\mathsf{U})$.

For $a,b \in \Sigma$ with $a \neq b$ we have

$$(\mathsf{X}_a \leqslant \mathsf{X}_b) = \bigvee_{c\in\Sigma} \big((\mathsf{X}_c \leqslant \mathsf{X}_a) \wedge (\mathsf{X}_c \leqslant \mathsf{X}_b) \wedge \mathsf{EX}(c \wedge \neg(\neg a \,\mathsf{U}\, b))\big).$$

Thus, it is enough to consider a conjunction $(\mathsf{X}_c \leqslant \mathsf{X}_a) \wedge \mathsf{EX}\, c$ with $a \neq c$. This is $\mathsf{EX}(c \wedge \mathsf{F}\, a) \wedge (a \vee \neg(\neg c \,\mathsf{U}\, a))$.

Next, for $a \in \Sigma$, we have

$$\mathsf{X}_a\,\varphi = (\neg a \wedge (\neg a\,\mathsf{U}\,(a \wedge \varphi))) \vee (a \wedge \mathsf{EX}(\neg a\,\mathsf{U}\,(a \wedge \varphi)))$$

and $\varphi\,\mathsf{U}_a\,\psi = (\neg a \vee \varphi)\,\mathsf{U}\,(a \wedge \psi)$. (This yields a direct proof for (1) $\subseteq$ (3) without the detour to process based logics.)

(2) $\subseteq$ (3): Let $i \in \mathcal{P}$. We have $\varphi\,\mathsf{U}_i\,\psi = (\neg\Sigma_i \vee \varphi)\,\mathsf{U}\,(\Sigma_i \wedge \psi)$ and

$$\mathsf{X}_i\,\varphi = \bigvee_{b\in\Sigma_i} \Big(\mathsf{X}_b\,\varphi \wedge \bigwedge_{a\in\Sigma_i\setminus\{b\}} \neg(\mathsf{X}_a \leqslant \mathsf{X}_b)\Big).$$

(3) $\subseteq$ (4): We have already seen that EX and U are expressible with SU. □

Remark 3. In the logic $\mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$, only the constants $(\mathsf{X}_a \leqslant \mathsf{X}_b)$ with $(a,b) \in D$ and $a \neq b$ are necessary. Indeed, we have $(\mathsf{X}_a \leqslant \mathsf{X}_a) = \mathsf{X}_a\,\top$ and we can replace $(\mathsf{X}_a \leqslant_1 \mathsf{X}_b)$ by $(\mathsf{X}_a \leqslant \mathsf{X}_b)$ with $(a,b) \in D$ and define $(\mathsf{X}_a \leqslant_n \mathsf{X}_b)$ for $n > 1$ inductively as in the proof of Proposition 2.

Remark 4. In Corollary 26 and 27 we will see that all the logics of Proposition 2 are expressively equivalent and correspond to the first order logic over traces. On the other hand, the logic $\mathrm{LocTL}_\Sigma[\mathsf{X}_a, \mathsf{U}_a]$ is strictly weaker. In fact, this fragment seems to be rather weak, even if we restrict ourselves to words over two letters. Assume that Σ contains two dependent letters b and c and let $\varphi \in \mathrm{LocTL}_\Sigma[X_a, U_a]$ be a formula of length n. Let $u = b(bc)^m$ and $v = (bc)^m$ with $m > n$ (possibly $m = \omega$). We can show that $u \models \varphi$ if and only if $v \models \varphi$. Since, $u \models (\mathsf{X}_b \leqslant \mathsf{X}_c)$ whereas $v \not\models (\mathsf{X}_b \leqslant \mathsf{X}_c)$. this shows that $\mathrm{LocTL}_\Sigma[\mathsf{X}_a, \mathsf{U}_a]$ is strictly weaker than $\mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$.

Note also that $\mathrm{LocTL}_\Sigma[X_a]$ is strictly weaker than $\mathrm{LocTL}_\Sigma[X_a, \mathsf{U}_a]$. Again, we assume that Σ contains two dependent letters b and c and we consider a formula $\varphi \in \mathrm{LocTL}_\Sigma[X_a]$ of length n. Then, for $m > n$ the traces (words) $(bc)^m$, $(bc)^m b$, $(bc)^m b^\omega$, and $(bc)^\omega$ are undistinguishable by φ. But, $(bc)^m \models \mathsf{F}(c \wedge \neg\,\mathsf{X}_b\,\top)$ whereas $(bc)^m b \not\models \mathsf{F}(c \wedge \neg\,\mathsf{X}_b\,\top)$, and $(bc)^m b^\omega \models \mathsf{F}\,\neg\,\mathsf{X}_c\,\top$ whereas $(bc)^\omega \not\models \mathsf{F}\,\neg\,\mathsf{X}_c\,\top$. Therefore, the fragment $\mathrm{LocTL}_\Sigma[X_a]$ is strictly weaker than $\mathrm{LocTL}_\Sigma[X_a, \mathsf{U}_a]$ both for finite and for infinite traces (or words).

Recall that a nonempty (finite or infinite) word w *initially satisfies* an LTL formula φ if $w, 1 \models \varphi$ where 1 is the first position in w. This can be extended directly to traces in $\mathbb{R}^1$ having a unique minimal vertex and we define $\mathcal{L}^1(\varphi) = \{t \in \mathbb{R}^1 \mid t \models \varphi\}$. This coincides with the classical notation for nonempty words.

The aim of the paper is to show that local temporal logics have the same expressive power as first order logic on traces. Since a first order logic formula can be evaluated on arbitrary traces in $\mathbb{R}$, we need to extend the *initial satisfiability* of local temporal logics to all traces in $\mathbb{R}$, not only to those having a unique minimal vertex. Two approaches have been used. In [6], an initial modality $\mathsf{EM}\,\varphi$ was introduced with the meaning $t \models \mathsf{EM}\,\varphi$ if there is a minimal position x in t with $t, x \models \varphi$. Then, an initial formula α is a Boolean combination of initial modalities. The local temporal logic based on EM, EX, and U is expressively complete if and only if the dependence alphabet (Σ, D) is a cograph [6]. Hence, in order to get a pure future expressively complete local temporal logic as aimed in the present paper, we cannot follow this strategy.

The other approach, which we adopt here, is to consider *rooted traces*. Let # be a new symbol, $\# \notin \Sigma$, and $t = [V, \leqslant, \lambda] \in \mathbb{R}(\Sigma, D)$. The rooted trace associated with t is $\#t$, where both # and t are viewed as traces over the alphabet $\Sigma' = \Sigma \cup \{\#\}$ together with the dependence relation $D' = D \cup (\{\#\} \times \Sigma) \cup (\Sigma \times \{\#\}) \cup \{(\#, \#)\}$. Thus we have introduced a unique minimal vertex, since # depends on every letter. In particular, $\#t \in \mathbb{R}^1(\Sigma', D')$. Then, for a formula in local temporal logic φ (over Σ), we define $\mathcal{L}(\varphi) = \mathcal{L}_\Sigma(\varphi) = \{t \in \mathbb{R}(\Sigma, D) \mid \#t \models \varphi\}$. Note that $\#\mathcal{L}_\Sigma(\varphi) = \mathcal{L}^1_{\Sigma'}(\varphi) \cap \#\mathbb{R}(\Sigma, D)$.

A formula $\varphi \in \mathrm{LocTL}_\Sigma[\cdots]$ is *insensitive to the minimal letter* (*iml* for short) if for all $t \in \mathbb{R}$ and $c \in \Sigma$ with $ct \in \mathbb{R}^1$ we have $\#t \models \varphi$ if and only if $ct \models \varphi$.

Lemma 5. *Let* $\varphi \in \mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$. *We can construct an iml formula* $\widehat{\varphi} \in \mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ *such that* $\mathcal{L}(\varphi) = \mathcal{L}(\widehat{\varphi})$.

Proof. We proceed by structural induction on φ. We have $\widehat{a} = \bot$ for each $a \in \Sigma$. Next, $\widehat{\mathsf{X}_a\,\varphi} = \mathsf{X}_a\,\varphi$ and $\widehat{(\mathsf{X}_a \leqslant \mathsf{X}_b)} = (\mathsf{X}_a \leqslant \mathsf{X}_b)$ since these formulae are already *iml*. Finally, $\varphi\,\widehat{\mathsf{U}_a}\,\psi = \mathsf{X}_a(\varphi\,\mathsf{U}_a\,\psi)$ for $a \in \Sigma$ since $\# \notin \Sigma$. □

4. Auxiliary constants

If there are letters $b, c \in \Sigma$ such that $\Uparrow x \cap \lambda^{-1}(b) \neq \emptyset$ and $\Uparrow x_b \cap \lambda^{-1}(c) \neq \emptyset$, we denote by $x_{bc} = (x_b)_c$ the minimal vertex of $\Uparrow x_b \cap \lambda^{-1}(c)$. We now define constants $(\mathsf{X}_{ac} = \mathsf{X}_{bc})$ for all $a, b, c \in \Sigma$ with $a \neq c \neq b$ by

$$t, x \models (\mathsf{X}_{ac} = \mathsf{X}_{bc}) \quad \text{if } x_{ac}, x_{bc} \text{ exist and } x_{ac} = x_{bc}.$$

It is far from being obvious that the new constants $(\mathsf{X}_{ac} = \mathsf{X}_{bc})$ can be expressed in $\mathrm{LocTL}_\Sigma[\mathsf{EX}, \mathsf{U}]$. We will devote the whole section to the proof of the following result, which is in view of Proposition 2, a priori, a stronger statement.

Proposition 6. *For all* $a, b, c \in \Sigma$ *with* $a \neq c \neq b$, *the constants* $(\mathsf{X}_{ac} = \mathsf{X}_{bc})$ *can be expressed in* $\mathrm{LocTL}_\Sigma[(\mathsf{X}_d \leqslant \mathsf{X}_e), \mathsf{X}_d, \mathsf{U}_d]$.

The remaining of this section is devoted to the technical proof of this proposition and can be skipped in a first reading.

The overall strategy is to proceed in $\mathcal{O}(n^3)$ rounds where $n = |\Sigma|$. In each round we introduce new formulae which are approximations of $(\mathsf{X}_{ac} = \mathsf{X}_{bc})$. At the end these approximations are getting so weak that we can replace them by *false*. In each round, when we replace an approximation we obtain a new formula of size $\mathcal{O}(n^2)$. Thus, overall $(\mathsf{X}_{ac} = \mathsf{X}_{bc})$ is replaced by a complex formula of exponential size in $|\Sigma|$.

Lemma 7. 1. *Let* z *be a vertex such that* $\lambda(z) = a$ *and* z_c *exists. There exist letters* $\{a_1, \ldots, a_{k-1}\} \subseteq \Sigma \setminus \{a, c\}$ *such that* $z < z_{a_1} < \cdots < z_{a_{k-1}} < z_c$ *and* $a = a_0 — a_1 — \cdots — a_{k-1} — a_k = c$ *in* (Σ, D).
2. *Let* x *be a vertex and* $\{a_1, \ldots, a_{k-1}\} \subseteq \Sigma \setminus \{a, c\}$ *such that* $x_a < x_{aa_1} < \cdots < x_{aa_{k-1}} < x_{ac}$ *and* $a = a_0 — a_1 — \cdots — a_{k-1} — a_k = c$ *in* (Σ, D). *If* $x_a \parallel x_c$, *then* $x_{aa_i} = x_{ca_i}$ *for some* $1 \leqslant i < k$.

Proof. 1. We use an induction on the size of the set $\{y \mid z \leqslant y < z_c\}$. Let y be a minimal vertex such that $z \leqslant y < z_c$ and $\lambda(y)$ depends on c. If $y = z$ then we have $a — c$ and we take $k = 1$. Assume now that $z < y$. By definition of y, we have $b = \lambda(y) \in \Sigma \setminus \{a, c\}$ and $y = z_b < z_c$. By induction, we find letters $\{a_1, \ldots, a_{k-2}\} \subseteq \Sigma \setminus \{a, b\}$ such that $z < z_{a_1} < \cdots < z_{a_{k-2}} < z_b$ and $a = a_0 — a_1 — \cdots — a_{k-2} — a_{k-1} = b$ in (Σ, D). We conclude easily since $z_b < z_c$, $b — c$, and $c \notin \{a_1, \ldots, a_{k-2}\}$ by definition of z_c.
2. Since $x_a \parallel x_c$, we have $(a, c) \in I$ and $k \geqslant 2$. The vertices x_c and $x_{aa_{k-1}}$ must be ordered. If $x_{aa_{k-1}} \leqslant x_c$ then $x_c = x_{ac}$, a contradiction. Hence, $x_c < x_{aa_{k-1}}$ and we can choose $0 < i < k$ minimal with $x_c < x_{aa_i}$. This implies $x_{ca_i} \leqslant x_{aa_i}$. We show that $x_{ca_i} = x_{aa_i}$. If $i = 1$ we let $y = x_a$ and if $i > 1$ we let $y = x_{aa_{i-1}}$. So, $(\lambda(y), a_i) \in D$ and y and x_{ca_i} are ordered. If $x_{ca_i} \leqslant y$ then $x_c < y$ and this excludes the case $i = 1$ since $x_a \parallel x_c$. Then, we get $x_c < x_{ca_i} \leqslant y = x_{aa_{i-1}}$ which contradicts the minimality of i. Therefore, $y < x_{ca_i}$ and using $y_{a_i} = x_{aa_i}$, we deduce that $x_{aa_i} \leqslant x_{ca_i}$ and therefore $x_{ca_i} = x_{aa_i}$. □

Let $a, c \in \Sigma$, $a \neq c$, and let $t \in \mathbb{R}$, $x \in t$ such that x_{ac} exists. Define $\delta_x(a, c)$ as the smallest integer $k \geqslant 1$ such that there exist letters $a_1, \cdots, a_{k-1}$ such that $x_a < x_{aa_1} < \cdots < x_{aa_{k-1}} < x_{ac}$ and $a = a_0 — a_1 — \cdots — a_{k-1} — a_k = c$ in (Σ, D). Note that such an integer k exists by Lemma 7 and $\delta_x(a, c) \leqslant |\Sigma| - 1$.

We also introduce the set $F_x(a, c)$ which consists of all pairs (d, e), $d \neq e$, such that either x_{de} does not exist or $x_{ac} < x_{de}$. Note that $|F_x(a, c)| \leqslant |\Sigma|^2 - |\Sigma|$. Throughout we use the following fact:

$$\text{if } x \leqslant y \text{ and } y_{fg} \leqslant x_{ac}, \text{ then } F_x(a, c) \subseteq F_y(f, g). \qquad (*)$$

This is trivial since if $x \leqslant y$ and y_{de} exists, then x_{de} exists and $x_{de} \leqslant y_{de}$. Moreover, if $x \leqslant y$ and $y_{fg} < x_{ac}$, then $F_x(a, c) \subsetneq F_y(f, g)$ since $(a, c) \in F_y(f, g)$ (even if y_{ac} does not exist).

Below we consider letters $a \neq c \neq b$ together with parameters $\delta_x(a, c) + \delta_x(b, c)$ and $|F_x(a, c)|$. We also introduce a flag $r \in \{0, 1\}$.

Proposition 8. *Let $a, b, c \in \Sigma$ with $a \neq c \neq b$. For each triple (m, ℓ, r) with $0 \leqslant m \leqslant |\Sigma|^2 - |\Sigma|$, $0 \leqslant \ell \leqslant 2|\Sigma| - 2$, and $r \in \{0, 1\}$ we can define a formula $(\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, r)$ in terms of $(\mathsf{X}_d < \mathsf{X}_e)$, X_d and U_d with $d, e \in \Sigma$ such that for all $x \in t \in \mathbb{R}$ the following assertions I and II are satisfied.*

I: If $t, x \models (\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, r)$, then $t, x \models (\mathsf{X}_{ac} = \mathsf{X}_{bc})$.
II: If the following four conditions $C_1, \ldots, C_4$ are simultaneously satisfied, then it holds: $t, x \models (\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, r)$.

C_1: $t, x \models (\mathsf{X}_{ac} = \mathsf{X}_{bc})$.
C_2: $|F_x(a, c)| = |F_x(b, c)| \geqslant m$.
C_3: $\delta_x(a, c) + \delta_x(b, c) \leqslant \ell$.
C_4: $r = 1$ or $t, x \models (\mathsf{X}_a \parallel \mathsf{X}_b) \wedge \neg[(\mathsf{X}_c < \mathsf{X}_a) \wedge (\mathsf{X}_c < \mathsf{X}_b)]$.

Corollary 9. *The formulae $(\mathsf{X}_{ac} = \mathsf{X}_{bc})$ and $(\mathsf{X}_{ac} = \mathsf{X}_{bc}, 0, 2|\Sigma| - 2, 1)$ are equivalent.*

Proof of Proposition 8. For $a = b$ we define $(\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, r)$ by the formula $\mathsf{X}_a \, \mathsf{X}_c \top$ which simply states that x_{ac} exists. Obviously, I and II are both satisfied for $a = b$. Hence in the following we may assume $|\{a, b, c\}| = 3$. Consider a triple (m, ℓ, r). If now either $m > |\Sigma|^2 - |\Sigma| - 2$ or $\ell \leqslant 1$, then we define $(\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, r)$ by *false*. Then I is trivially true. Assertion II also holds since if x_{ac} and x_{bc} exist and $x_{ac} = x_{bc}$ then either C_2 (for $m > |\Sigma|^2 - |\Sigma| - 2$) or C_3 (for $\ell \leqslant 1$) is impossible.

In the following we may assume by induction that formulae are defined satisfying both I and II for all triples (m', ℓ', r') where either $m' > m$ or $m' = m, \ell' < \ell$ or $m' = m, \ell' = \ell$, and $r' < r$.

Case $r = 1$: We define $(\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, 1)$ by $\varphi_1 \vee \varphi_2 \vee \varphi_3$ where:

$$\begin{aligned}
\varphi_1 &= ((\mathsf{X}_a < \mathsf{X}_b) \wedge \mathsf{X}_a(\mathsf{X}_b < \mathsf{X}_c)) \vee ((\mathsf{X}_b < \mathsf{X}_a) \wedge \mathsf{X}_b(\mathsf{X}_a < \mathsf{X}_c)),\\
\varphi_2 &= (\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, 0),\\
\varphi_3 &= (\mathsf{X}_a \parallel \mathsf{X}_b) \wedge \psi_1 \wedge \psi_2,\\
\psi_1 &= (\mathsf{X}_c < \mathsf{X}_a) \wedge (\mathsf{X}_c < \mathsf{X}_b),\\
\psi_2 &= \psi_1 \,\mathsf{U}_c\, ((\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, 0) \wedge \neg\psi_1).
\end{aligned}$$

First, we show assertion I: Let $t, x \models (\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, 1)$. If $t, x \models \varphi_1$, then $t, x \models (\mathsf{X}_{ac} = \mathsf{X}_{bc})$ by a direct verification. For $t, x \models \varphi_2$ we obtain the implication by induction. Hence let $t, x \models \varphi_3$. Choose a vertex $y \in t$ which is maximal with respect to the three properties $\lambda(y) = c$, $x < y < x_a$, and $x < y < x_b$. This vertex exists since $t, x \models \psi_1$. In particular $y_a = x_a$ and $y_b = x_b$ and by maximality of y we get $t, y \models \neg\psi_1$. Hence $t, y \models (\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, 0)$ since $t, x \models \psi_2$. It follows by induction that $x_{ac} = y_{ac} = y_{bc} = x_{bc}$ as desired.

Now we show II for $r = 1$. Condition C_1 says that x_{ac} and x_{bc} exist and that we have $x_{ac} = x_{bc}$. If $x_a < x_b$ or $x_b < x_a$, then $t, x \models \varphi_1$. Since $a \neq b$ we may therefore assume that $t, x \models (\mathsf{X}_a \parallel \mathsf{X}_b)$. If now in addition $t, x \models \neg\psi_1$, then $C_1, \ldots, C_4$ hold for the triple $(m, \ell, 0)$ as well. We obtain $t, x \models \varphi_2$ by induction and we are done in this case. Hence we may assume both $x_c < x_a$ and $x_c < x_b$. Now, again choose $y \in t$ maximal with respect to $\lambda(y) = c$, $x < y < x_a$, and $x < y < x_b$. Clearly, $t, y \models \neg\psi_1$ by maximality of y. In order to show that $t, x \models \varphi_3$ it is enough to verify $t, y \models (\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, 0)$. By induction, this requires to check $C_1, \ldots, C_4$ for y. We have $y_a = x_a$, $y_b = x_b$, $y_{ac} = x_{ac}$ and $y_{bc} = x_{bc}$, so the two conditions C_1 and C_4 are true. Clearly $F_y(a, c) = F_y(b, c)$, because $y_{ac} = y_{bc}$. Moreover $F_x(a, c) \subseteq F_y(a, c)$ by $(*)$ hence C_2 holds. Finally, $\delta_y(a, c) = \delta_x(a, c)$ and $\delta_y(b, c) = \delta_x(b, c)$, because $y_a = x_a$ and $y_b = x_b$. Hence C_3 holds, too.

Case $r = 0$: We define $(\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, 0)$ by $\tau_0 \vee \tau_1 \vee \tau_2 \vee \tau_3$ where:

$$\begin{aligned}
\tau_0 &= (\mathsf{X}_a < \mathsf{X}_c) \wedge (\mathsf{X}_b < \mathsf{X}_c),\\
\tau_1 &= (\mathsf{X}_c < \mathsf{X}_a) \wedge \bigvee_{b \neq b' \neq c} \tau(b, b') \wedge \mathsf{X}_c(\mathsf{X}_{ac} = \mathsf{X}_{b'c}, m, \ell - 1, 1),\\
\tau_2 &= (\mathsf{X}_c < \mathsf{X}_b) \wedge \bigvee_{a \neq a' \neq c} \tau(a, a') \wedge \mathsf{X}_c(\mathsf{X}_{a'c} = \mathsf{X}_{bc}, m, \ell - 1, 1),\\
\tau_3 &= \bigvee_{\substack{a \neq a' \neq c \\ b \neq b' \neq c}} \tau(a, a') \wedge \tau(b, b') \wedge \mathsf{X}_c(\mathsf{X}_{a'c} = \mathsf{X}_{b'c}, m, \ell - 2, 1),\\
\tau(a, a') &= (\mathsf{X}_{aa'} = \mathsf{X}_{ca'}, m + 2, 2|\Sigma| - 2, 1) \wedge \mathsf{X}_a(\mathsf{X}_{a'} < \mathsf{X}_c),\\
\tau(b, b') &= (\mathsf{X}_{bb'} = \mathsf{X}_{cb'}, m + 2, 2|\Sigma| - 2, 1) \wedge \mathsf{X}_b(\mathsf{X}_{b'} < \mathsf{X}_c).
\end{aligned}$$

To see assertion I first, suppose $t,x \models (\mathsf{X}_{ac} = \mathsf{X}_{bc}, m, \ell, 0)$. If $t,x \models \tau_0$, then x_{ac}, x_{bc} exist and moreover, $x_c = x_{ac} = x_{bc}$ in this case. In particular, $t,x \models (\mathsf{X}_{ac} = \mathsf{X}_{bc})$. The following arguments are quite similar for τ_1, τ_2 and τ_3. The most elaborate one is for τ_3. So, we treat only this case and we assume that $t,x \models \tau_3$. Therefore we find $a \neq a' \neq c$ and $b \neq b' \neq c$ such that the following statements hold where we define $y = x_c$:

$$\begin{aligned} & x_a < x_{aa'} = x_{ca'} = y_{a'} < x_{ac}, && \text{(by induction and } \tau(a,a')) \\ & x_b < x_{bb'} = x_{cb'} = y_{b'} < x_{bc}, && \text{(by induction and } \tau(b,b')) \\ & y_{a'c} = y_{b'c}. && \text{(by induction and } \tau_3\text{, last part)} \end{aligned}$$

We conclude $x_{ac} = y_{a'c} = y_{b'c} = x_{bc}$ and hence $t,x \models (\mathsf{X}_{ac} = \mathsf{X}_{bc})$ as desired.

We still have to verify assertion II for $r = 0$ and $|\{a,b,c\}| = 3$. Consider $x \in t \in \mathbb{R}$ such that $C_1, \ldots, C_4$ are all satisfied. In particular, x_{ac}, x_{bc} exist and we have $x_{ac} = x_{bc}$. If $x_a < x_c$ then $x_c = x_{ac} = x_{bc}$ hence also $x_b < x_c$ and $t,x \models \tau_0$. Similarly, if $x_b < x_c$ then $t,x \models \tau_0$. Hence in the following we assume that neither $x_a < x_c$ nor $x_b < x_c$.
There are three cases:

(1) $x_c < x_a$,
(2) $x_c < x_b$ and
(3) neither $x_c < x_a$ nor $x_c < x_b$.

These cases correspond to τ_1, τ_2, and τ_3, respectively. Since $r = 0$, C_4 implies $x_a \parallel x_b$ and $\neg(x_c < x_a \wedge x_c < x_b)$. Hence, in case 1, using $\neg(x_b < x_c)$ and $b \neq c$, we get $x_b \parallel x_c$. Similarly, in case 2 we have $x_a \parallel x_c$ and in case 3 we have both $x_a \parallel x_c$ and $x_b \parallel x_c$. So in all cases we have at least two concurrent vertices and we will apply the following

Claim 10. *If $x_a \parallel x_c$ then we find $a' \in \Sigma \setminus \{a,c\}$ such that both $\delta_{x_c}(a',c) < \delta_x(a,c)$ and $t,x \models \tau(a,a')$.*

Let $k = \delta_x(a,c)$, by definition we find letters $a_1, \ldots, a_{k-1} \subseteq \Sigma \setminus \{a,c\}$ such that $x_a < x_{aa_1} < \cdots < x_{aa_{k-1}} < x_{ac}$ and $a = a_0 — a_1 — \cdots — a_{k-1} — a_k = c$ in (Σ, D). Since $x_a \parallel x_c$, we may apply Lemma 7 (2) and we find $1 \leqslant i < k$ with $x_{aa_i} = x_{ca_i}$. Let $a' = a_i$ and $y = x_c$. For $i < j \leqslant k$ we have $y_{a'a_j} = x_{aa_j}$. Hence, $\delta_y(a',c) \leqslant k - i < \delta_x(a,c)$.

To see the claim it remains to show that $t,x \models \tau(a,a')$. Since $x_{aa'} < x_{ac}$ we have to show $t,x \models (\mathsf{X}_{aa'} = \mathsf{X}_{ca'}, m+2, 2|\Sigma|-2, 1)$. Let us consider conditions $C_1, \ldots, C_4$ with respect to (a,c,a') and the triple $(m+2, 2|\Sigma|-2, 1)$. Condition C_1 holds since $x_{aa'} = x_{ca'}$. Condition C_3 trivially holds since $\delta_x(a,a') + \delta_x(c,a') \leqslant 2|\Sigma| - 2$. Condition C_4 trivially holds since $r = 1$. Thus, we need to verify C_2, only. Since $x_{aa'} = x_{ca'}$, we have $F_x(a,a') = F_x(c,a')$. Since $x_{aa'} < x_{ac}$ we obtain $F_x(a,c) \subseteq F_x(a,a')$ and in fact $(a,c), (b,c) \in F_x(a,a') \setminus F_x(a,c)$. Hence $|F_x(a,a')| = |F_x(c,a')| \geqslant m+2$. Thus all four conditions are satisfied and using the induction hypothesis we get $t,x \models (\mathsf{X}_{aa'} = \mathsf{X}_{ca'}, m+2, 2|\Sigma|-2, 1)$ which concludes the proof of the claim. □

We come back to the proof of the three cases. We start with case (2). We have $x_c < x_b$ and $x_a \parallel x_c$. Let a' be given by Claim 10, and let $y = x_c$. We show that $C_1, \ldots, C_4$ hold for y, (a',b,c) and $(m, \ell-1, 1)$. We have $x_a < x_{aa'} = x_{ca'} = y_{a'} < x_{ac}$, hence $y_{a'c} = x_{ac}$. Also, $x < y < x_b$ implies $y_{bc} = x_{bc}$. Therefore, $y_{a'c} = x_{ac} = x_{bc} = y_{bc}$ and C_1 holds. Using $(*)$, we get $F_x(a,c) \subseteq F_y(a',c)$ and

C_2 holds. Claim 10 also implies C_3 since $\delta_y(a',c) < \delta_x(a,c)$ and $\delta_y(b,c) = \delta_x(b,c)$. Finally, C_4 trivially holds since $r = 1$. By induction, we get $t, y \models (\mathsf{X}_{a'c} = \mathsf{X}_{bc}, m, \ell - 1, 1)$ and therefore, $t,x \models \tau_2$.

Case (1) is symmetrical. For case (3), we apply twice Claim 10 in order to get a' and b'. We show that $C_1,\ldots,C_4$ hold for $y = x_c$, (a', b', c) and $(m, \ell - 2, 1)$. As above, we have $y_{a'c} = x_{ac}$ and $y_{b'c} = x_{bc}$, hence C_1 holds. From Claim 10 we get $\delta_y(a',c) + \delta_y(b',c) \leqslant \delta_x(a,c) + \delta_x(b,c) - 2 \leqslant \ell - 2$ and C_3 holds. Finally, C_4 trivially holds since $r = 1$ and C_2 can be deduced using $(*)$ as above. By induction, we get $t, y \models (\mathsf{X}_{a'c} = \mathsf{X}_{b'c}, m, \ell - 2, 1)$ and therefore, $t,x \models \tau_3$. □

5. Lifting Theorem

In this section A denotes a subset of Σ. For $x \in t \in \mathbb{R}$ we define $\mu_A(x,t)$ to be the prefix of $\uparrow x$ which is given by the set of vertices

$$\{z \in t \mid x \leqslant z \text{ and } \forall y, x < y \leqslant z \Rightarrow \lambda(y) \in A\}.$$

Thus, we always have $x \in \mu_A(x,t)$ and all other vertices of $\mu_A(x,t)$ have a label in A. Indeed, $\mu_A(x,t)$ is the maximal prefix of $\uparrow x$ having this property. The aim of this section is to establish the following theorem. The proof relies substantially on Proposition 6.

Theorem 11 (*Lifting*). *Let $\varphi \in \mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ and $A \subseteq \Sigma$. Then we effectively find a formula $\overline{\varphi}^A \in \mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ such that for all $x \in t \in \mathbb{R}$ we have*

$$\mu_A(x,t), x \models \varphi \quad \text{if and only if} \quad t,x \models \overline{\varphi}^A. \tag{1}$$

The rest of this section is devoted to the proof of this theorem, which is done by structural induction on φ. We start with the following observations: $\overline{a}^A = a$ for all $a \in \Sigma$, $\overline{\varphi \wedge \psi}^A = \overline{\varphi}^A \wedge \overline{\psi}^A$, and $\overline{\neg\varphi}^A = \neg\,\overline{\varphi}^A$.

Now, $\mu_A(x,t), x \models (\mathsf{X}_a \leqslant \mathsf{X}_b)$ if and only if both $t,x \models (\mathsf{X}_a \leqslant \mathsf{X}_b)$ and $x_b \in \mu_A(x,t)$. However, $x_b \in \mu_A(x,t)$ can be expressed using the next lemma, the proof of which is easy and omitted.

Lemma 12. *For $A \subseteq \Sigma$ and $a \in \Sigma$, we let $\xi_1(A,a) = \mathsf{X}_a \top \wedge \bigwedge_{c \notin A} \neg(\mathsf{X}_c \leqslant \mathsf{X}_a)$. Let $t \in \mathbb{R}$ and $x \in t$. Then,*

$$x_a \textit{ exists and } x_a \in \mu_A(x,t) \quad \textit{if and only if} \quad t,x \models \xi_1(A,a).$$

The remaining cases, $\overline{\mathsf{X}_a \varphi}^A$ and $\overline{\varphi \,\mathsf{U}_a\, \psi}^A$, are much more involved. We introduce first another macro $\mathrm{Switch}_{A,B,a}$ for $a \in B \subseteq A$. We want that $t,x \models \mathrm{Switch}_{A,B,a}$ implies that both $x_a \in \mu_A(x,t)$ exists and $\mu_A(x,t) \cap \uparrow x_a = \mu_B(x_a,t)$. Moreover, whenever $x_a \in \mu_A(x,t)$ exists, then we want that $t,x \models \mathrm{Switch}_{A,B,a}$ for some $a \in B \subseteq A$. This will be stated in Proposition 14 formally. The construction of the macro $\mathrm{Switch}_{A,B,a}$ is based on the next lemma.

Lemma 13. *Let $x \in t \in \mathbb{R}$ and $a \in \Sigma$ such that x_a exists and $x_a \in \mu_A(x,t)$. Define*

$$B = \{a\} \cup \{b \in A \setminus \{a\} \mid t,x \models \bigwedge_{c \notin A} \neg(\mathsf{X}_{ab} = \mathsf{X}_{cb})\}.$$

Then we have $a \in B \subseteq A$ *and* $\mu_A(x,t) \cap \uparrow x_a = \mu_B(x_a,t)$.

Proof. Observe that $x_a \in \mu_A(x,t)$ implies $a \in A$, hence $B \subseteq A$.

For $\mu_A(x,t) \cap \uparrow x_a \subseteq \mu_B(x_a,t)$ consider $z \in \mu_A(x,t) \cap \uparrow x_a$ and $x_a < y \leqslant z$. We have to show that $b = \lambda(y) \in B$. Since $x < y \leqslant z$ and $z \in \mu_A(x,t)$ we have $b \in A$. If $b = a$, then $b \in B$. Assume now that $b \neq a$ so that $b \in A \setminus \{a\}$ and let $c \in \Sigma$ be such that $x_{ab} = x_{cb}$. We have $x < x_c < x_{cb} = x_{ab} \leqslant y \leqslant z$ and using $z \in \mu_A(x,t)$ we get $c \in A$. Therefore, $b \in B$.

For the other direction, let $z \in \mu_B(x_a,t)$. We have to prove that $z \in \mu_A(x,t)$. For this, it is enough to show that $x_c \leqslant z$ implies $c \in A$ for each $c \in \Sigma$. So let $c \in \Sigma$ be such that x_c exists and $x_c \leqslant z$. If $x_c \leqslant x_a$ then $c \in A$ since $x_a \in \mu_A(x,t)$. If $x_a < x_c$ then $c \in B \subseteq A$ since $z \in \mu_B(x_a,t)$. Hence we assume in the following $x_a \parallel x_c$. Now choose $y \in t$ which is minimal with respect to the properties $x_a \leqslant y \leqslant z$ and $x_c \leqslant y \leqslant z$. Since $x_a \parallel x_c$, we obtain $x_a < y$ and $x_c < y$. Let $b = \lambda(y)$. We show that $y = x_{ab} = x_{cb}$. Without loss of generality, we assume that $x_{ab} \leqslant x_{cb}$ and we consider y' with $x_c \leqslant y' \lessdot y$. We have $(b,\lambda(y')) \in D$ hence x_{ab} and y' must be ordered. Using the minimality of y we deduce that $x_{ab} \leqslant y'$ is impossible. Hence, $y' < x_{ab} \leqslant x_{cb} \leqslant y$ and using $y' \lessdot y$ we get $y = x_{ab} = x_{cb}$ as desired. Now, $x_a < y \leqslant z \in \mu_B(x_a,t)$ implies $b \in B$. Also, $b = a$ is not possible since otherwise x_a and y' must be ordered, but $y' \leqslant x_a$ contradicts $x_a \parallel x_c$ and $x_a < y'$ contradicts the minimality of y. Therefore $b \in B \setminus \{a\}$ and since $x_{ab} = x_{cb}$ we must have $c \in A$ as required. □

Let $a \in \Sigma$ and $A, B \subseteq \Sigma$. If $a \notin B$ or $B \not\subseteq A$ then we define $\mathrm{Switch}_{A,B,a} = \bot$. If, on the other hand, $a \in B \subseteq A$ then we define $\mathrm{Switch}_{A,B,a}$ as a conjunction $\xi_1(A,a) \wedge \xi_2(A,B,a) \wedge \xi_3(A,B,a)$ where

$$\xi_2(A,B,a) = \bigwedge_{b \in B \setminus \{a\}} \bigwedge_{c \notin A} \neg(\mathsf{X}_{ab} = \mathsf{X}_{cb}),$$

$$\xi_3(A,B,a) = \bigwedge_{b \in A \setminus B} \bigvee_{c \notin A} (\mathsf{X}_{ab} = \mathsf{X}_{cb}).$$

Note that $\mathrm{Switch}_{A,B,a}$ is in $\mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ by Proposition 6. As a consequence of Lemmata 12 and 13 we obtain the following proposition.

Proposition 14.

(1) *Let* $a \in \Sigma$ *and* $A, B \subseteq \Sigma$. *If* $t, x \models \mathrm{Switch}_{A,B,a}$ *then* $a \in B \subseteq A$, x_a *exists*, $x_a \in \mu_A(x,t)$ *and* $\mu_A(x,t) \cap \uparrow x_a = \mu_B(x_a,t)$.

(2) *Let* $a \in \Sigma$ *and* $A \subseteq \Sigma$. *If* x_a *exists and* $x_a \in \mu_A(x,t)$ *then we have* $t, x \models \mathrm{Switch}_{A,B,a}$ *for some* $a \in B \subseteq A$.

Proof. If $t, x \models \mathrm{Switch}_{A,B,a}$ then x_a exists and $x_a \in \mu_A(x,t)$ by Lemma 12. Moreover, B is exactly the set as defined in Lemma 13, since $a \in B$. Hence we obtain (1) by Lemma 13.

Assume now that x_a exists and $x_a \in \mu_A(x,t)$. We get $t \models \xi_1$ by Lemma 12. Let B be defined as in Lemma 13. We obtain $t \models \xi_2(A,B,a) \wedge \xi_3(A,B,a)$. □

We can now easily deal with the case $\mathsf{X}_a\,\varphi$ in the inductive proof of Theorem 11.

Lemma 15. *The formula* $\overline{\mathsf{X}_a\,\varphi}^A = \bigvee_B \mathrm{Switch}_{A,B,a} \wedge \mathsf{X}_a\ \overline{\varphi}^B$ *satisfies Eq.* (*1*).

Proof. Assume first that $\mu_A(x,t),x \models \mathsf{X}_a\,\varphi$. Then, x_a exists, $x_a \in \mu_A(x,t)$ and $\mu_A(x,t),x_a \models \varphi$. By Proposition 14(2), we have $t,x \models \mathrm{Switch}_{A,B,a}$ for some $a \in B \subseteq A$. We get $\mu_A(x,t) \cap \uparrow x_a = \mu_B(x_a,t)$ by Proposition 14(1) and since the evaluation of a formula only depends on the future of the current vertex we get $\mu_B(x_a,t),x_a \models \varphi$. By structural induction we obtain $t,x_a \models \overline{\varphi}^B$ and therefore $t,x \models \mathsf{X}_a\ \overline{\varphi}^B$.

Conversely, assume that $t,x \models \mathrm{Switch}_{A,B,a} \wedge \mathsf{X}_a\ \overline{\varphi}^B$ for some $B \subseteq \Sigma$. Then, $a \in B \subseteq A$, x_a exists, $x_a \in \mu_A(x,t)$ and $\mu_A(x,t) \cap \uparrow x_a = \mu_B(x_a,t)$ by Proposition 14(1). Using $t,x_a \models \overline{\varphi}^B$ we get by structural induction that $\mu_B(x_a,t),x_a \models \varphi$. It follows $\mu_A(x,t),x_a \models \varphi$ and $\mu_A(x,t),x \models \mathsf{X}_a\,\varphi$. □

For the remaining case $\overline{\varphi\,\mathsf{U}_a\,\psi}^A$ of the proof of Theorem 11, we also use an induction on A. Note first that $\neg a \wedge \varphi\,\mathsf{U}_a\,\psi = \neg a \wedge \mathsf{X}_a(a \wedge \varphi\,\mathsf{U}_a\,\psi)$, hence it is enough to lift a conjunction $a \wedge \varphi\,\mathsf{U}_a\,\psi$. We have $\overline{a \wedge \varphi\,\mathsf{U}_a\,\psi}^{\emptyset} = a \wedge \overline{\psi}^{\emptyset}$. Now, we may assume that $\overline{a \wedge \varphi\,\mathsf{U}_a\,\psi}^B$ is already defined for all $B \subsetneq A$ and we can use the following lemma.

Lemma 16. *The formula*

$$\overline{a \wedge \varphi\,\mathsf{U}_a\,\psi}^A = a \wedge (\mathrm{Switch}_{A,A,a} \wedge\ \overline{\varphi}^A)\ \mathsf{U}_a\ (\overline{\psi}^A \vee (\overline{\varphi}^A \wedge \sigma)),$$

where

$$\sigma = \bigvee_{B \subsetneq A} \mathrm{Switch}_{A,B,a} \wedge \mathsf{X}_a\ \overline{a \wedge \varphi\,\mathsf{U}_a\,\psi}^B$$

satisfies Eq. (*1*).

Proof. Assume first that $\mu_A(x,t),x \models a \wedge \varphi\,\mathsf{U}_a\,\psi$ and consider a chain $x = x_0 < x_1 < \cdots < x_k$ with $k \geqslant 0$ such that $x_{i+1} = (x_i)_a$, $\mu_A(x,t),x_i \models \varphi$ for $0 \leqslant i < k$ and $\mu_A(x,t),x_k \models \psi$. Choose $j \in \{0,\ldots,k\}$ maximal such that $t,x_i \models \mathrm{Switch}_{A,A,a}$ for all $0 \leqslant i < j$. Then we have $\mu_A(x,t) \cap \uparrow x_i = \mu_A(x_i,t)$ for all $0 \leqslant i \leqslant j$ by Proposition 14(1). (Note that both indices 0 and j are included as a possible value for i.) Hence $\mu_A(x_i,t),x_i \models \varphi$ for $0 \leqslant i < j$ and by structural induction we get $t,x_i \models \overline{\varphi}^A$ for $0 \leqslant i < j$.

Now, if $j = k$ then we get $\mu_A(x_k,t),x_k \models \psi$ and by structural induction $t,x_k \models \overline{\psi}^A$. Therefore, $t,x \models \overline{a \wedge \varphi\,\mathsf{U}_a\,\psi}^A$. On the other hand, if $j < k$ then we have $\mu_A(x_j,t),x_j \models \varphi$ and also $\mu_A(x_j,t),x_j \models \mathsf{X}_a(a \wedge \varphi\,\mathsf{U}_a\,\psi)$. By structural induction we deduce that $t,x_j \models \overline{\varphi}^A$. Now, since $t,x_j \not\models \mathrm{Switch}_{A,A,a}$ and $x_{j+1} = (x_j)_a \in \mu_A(x,t) \cap \uparrow x_j = \mu_A(x_j,t)$ exists, we have $t,x_j \models \mathrm{Switch}_{A,B,a}$ for some $a \in B \subsetneq A$ by Proposition 14(2). Hence, arguing as above, we deduce that $\mu_B(x_{j+1},t),x_{j+1} \models a \wedge \varphi\,\mathsf{U}_a\,\psi$. Since $B \subsetneq A$ we get $t,x_{j+1} \models \overline{a \wedge \varphi\,\mathsf{U}_a\,\psi}^B$ by induction on A. We deduce that $t,x_j \models \sigma$, and hence $t,x \models \overline{a \wedge \varphi\,\mathsf{U}_a\,\psi}^A$.

Conversely, assume $t,x \models a \wedge (\mathrm{Switch}_{A,A,a} \wedge \overline{\varphi}^A)\ \mathsf{U}_a\ (\overline{\psi}^A \vee (\overline{\varphi}^A \wedge \sigma))$. This means that for some $j \geqslant 0$ there is a chain $x = x_0 < x_1 < \cdots < x_j$ such that we have $x_{i+1} = (x_i)_a$ and $t,x_i \models \mathrm{Switch}_{A,A,a} \wedge \overline{\varphi}^A$ for $0 \leqslant i < j$ and $t,x_j \models \overline{\psi}^A \vee (\overline{\varphi}^A \wedge \sigma)$. By structural induction we obtain either $\mu_A(x_i,t),x_i \models \varphi$ for $0 \leqslant i < j$ and $\mu_A(x_j,t),x_j \models \psi$, or $\mu_A(x_i,t),x_i \models \varphi$ for $0 \leqslant i \leqslant j$ and $t,x_j \models \sigma$. Using Proposition

14(1), we obtain by induction on i that $\mu_A(x,t) \cap \uparrow x_i = \mu_A(x_i,t)$ for $0 \leqslant i \leqslant j$. Hence, we get either $\mu_A(x,t),x_i \models \varphi$ for $0 \leqslant i < j$ and $\mu_A(x,t),x_j \models \psi$, or $\mu_A(x,t),x_i \models \varphi$ for $0 \leqslant i \leqslant j$ and $t,x_j \models \sigma$. The first case means $\mu_A(x,t),x \models \varphi \mathsf{U}_a \psi$ as desired. Assume now that we are in the second case. For some $B \subsetneq A$ we have $t,x_j \models \mathrm{Switch}_{A,B,a} \wedge \mathsf{X}_a \overline{a \wedge \varphi \mathsf{U}_a \psi}^B$. Let $y = x_j$ so that $t, y_a \models \overline{\varphi \mathsf{U}_a \psi}^B$. Since $B \subsetneq A$ we obtain $\mu_B(y_a,t), y_a \models \varphi \mathsf{U}_a \psi$ by induction. Using Proposition 14(1) we know that $\mu_B(y_a,t) = \mu_A(y,t) \cap \uparrow y_a$. Since also $\mu_A(y,t) = \mu_A(x,t) \cap \uparrow y$ we obtain $\mu_B(y_a,t) = \mu_A(x,t) \cap \uparrow y_a$. Therefore, $\mu_A(x,t), y_a \models \varphi \mathsf{U}_a \psi$ and since $\mu_A(x,t),x_i \models \varphi$ for $0 \leqslant i \leqslant j$ we get again $\mu_A(x,t),x \models \varphi \mathsf{U}_a \psi$. □

6. Expressive completeness

The aim here is to establish the following result.

Theorem 17. *Let $L \subseteq \mathbb{R}$ be expressible in the first order logic* $\mathrm{FO}_\Sigma(<)$. *Then we can construct* $\varphi \in \mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ *such that* $L = \mathcal{L}(\varphi)$.

From the semantics of SU, it is classical (and easy to see) that with any formula $\varphi \in \mathrm{LocTL}_\Sigma(\mathsf{SU})$ we can associate a first order formula $\widetilde{\varphi}$ with one free variable such that for any $t \in \mathbb{R}$ and $x \in t$, we have $t,x \models \varphi$ if and only if $t \models \widetilde{\varphi}(x)$. Moreover, it is enough to use at most 3 distinct variable names for $\widetilde{\varphi}(x)$. Thus, we obtain as a direct consequence of Theorem 17 and Proposition 2:

Corollary 18 ([36,15]). *Let $L \subseteq \mathbb{R}$ be expressible in the first order logic* $\mathrm{FO}_\Sigma(<)$ *then it is expressible in* $\mathrm{FO}^3_\Sigma(<)$, *where* $\mathrm{FO}^3_\Sigma(<)$ *is the subset of first order formulae using at most 3 distinct variables.*

For the proof of Theorem 17 we use the algebraic notion of *recognizability* and the notion of *aperiodic* languages. Recognizability is defined as follows. Let $h : \mathbb{M} \to M$ be a morphism to a finite monoid M. For $s,t \in \mathbb{R}$, we say that s and t are h-similar, denoted by $s \sim_h t$, if we can write $s = \prod_{0 \leqslant i < n} s_i$ and $t = \prod_{0 \leqslant i < n} t_i$ with $s_i, t_i \in \mathbb{M} \setminus \{1\}$ and $h(s_i) = h(t_i)$ for all $0 \leqslant i < n$, where $n \in \mathbb{N} \cup \{\omega\}$. The transitive closure $\approx_h$ of $\sim_h$ is an equivalence relation. For $t \in \mathbb{R}$, we denote by $[t]_h$ the equivalence class of t under $\approx_h$. In case that there is no ambiguity, we simply write $[t]$, $\approx$, and $\sim$. Note that there are three cases: an equivalence class is either reduced to the empty trace ($[t] = \{1\}$), or consists of finite non-empty traces only ($[t] \subseteq \mathbb{M} \setminus \{1\}$), or consists of infinite traces only ($[t] \subseteq \mathbb{R} \setminus \mathbb{M}$). Since M is finite, the equivalence relation $\approx_h$ is of finite index with at most $1 + |M| + |M|^2$ equivalence classes. This fact is well-known and can be derived by some standard Ramsey argument, see e.g. [17]. A trace language $L \subseteq \mathbb{R}$ is *recognized* by h, if $t \in L$ implies $[t]_h \subseteq L$ for all $t \in \mathbb{R}$. This means that L is saturated by $\approx_h$ (or equivalently by $\sim_h$).

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

Theorem 19 ([10,11]). *A language $L \subseteq \mathbb{R}$ is expressible in* $\mathrm{FO}_\Sigma(<)$ *if and only if it is an aperiodic language.*

Theorem 17 is a direct consequence of the *only-if* direction of Theorem 19 and the following result. Theorem 20 is in fact our main technical contribution. We give a self-contained proof for it.

Theorem 20. *Let $L \subseteq \mathbb{R}$ be an aperiodic language. Then we can construct a formula $\varphi \in \mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ such that $\mathcal{L}(\varphi) = L$.*

Recall that for $A \subseteq \Sigma$, we denote by $D(A)$ the set of letters that depend on some letter in A. A morphism h is called *weakly alphabetic*, if $h(r) = h(s)$ implies $D(\mathrm{alph}(r)) = D(\mathrm{alph}(s))$ for all $r, s \in \mathbb{M} \setminus \{1\}$. Note that this condition is trivially satisfied for free monoids: If $\mathbb{M}$ is free, then all morphisms are weakly alphabetic.

The power set $(2^\Sigma, \cup)$ is an aperiodic monoid and the mapping $\mathbb{M} \to 2^\Sigma, t \mapsto D(\mathrm{alph}(t))$ is a weakly alphabetic morphism. It follows that every aperiodic language $L \subseteq \mathbb{R}$ can be recognized by some weakly alphabetic morphism, because we may replace a morphism $g : \mathbb{M} \to M$ by $h : \mathbb{M} \to M \times 2^\Sigma$ with $h(t) = (g(t), D(\mathrm{alph}(t))$. Of course, h is weakly alphabetic, recognizes L if g does, and $M \times 2^\Sigma$ is finite and aperiodic, if M shares this property.

Remark 21. Let $h : \mathbb{M} \to M$ be a weakly alphabetic morphism and let s, s', t, t' be such that $[s] = [s']$, and $[t] = [t']$. Then, $D(\mathrm{alphinf}(s)) = D(\mathrm{alphinf}(s'))$ and $D(\mathrm{alph}(t)) = D(\mathrm{alph}(t'))$. Hence also $\mathrm{alphinf}(s) \times \mathrm{alph}(t) \subseteq I$ if and only if $\mathrm{alphinf}(s') \times \mathrm{alph}(t') \subseteq I$. To show this last statement consider $A, A', B, B' \subseteq \Sigma$ such that $D(A) = D(A'), D(B) = D(B')$ and $A \times B \subseteq I$. Let $(a', b') \in A' \times B'$ and assume that $(a', b') \in D$. Then $a' \in D(B') = D(B)$ and we find $b \in B$ with $(a', b) \in D$. Now, $b \in D(A') = D(A)$ and we find $a \in A$ with $(a, b) \in D$, a contradiction. Therefore, $A' \times B' \subseteq I$.

Lemma 22. *Let $h : \mathbb{M} \to M$ be a weakly alphabetic morphism and $s, t \in \mathbb{R}$ such that $\mathrm{alphinf}(s) \times \mathrm{alph}(t) \subseteq I$. Then we have $[s][t] \subseteq [st]$.*

Proof. Let $s \sim s'$ and $t \sim t'$. We find $n \in \mathbb{N} \cup \{\omega\}$ and factorizations $s = \prod_{0 \leqslant i < n} s_i$, $s' = \prod_{0 \leqslant i < n} s'_i$, $t = \prod_{0 \leqslant i < n} t_i$ and $t' = \prod_{0 \leqslant i < n} t'_i$ such that $h(s_i) = h(s'_i), h(t_i) = h(t'_i), s_i t_i \neq 1 \neq s'_i t'_i$ for all $0 \leqslant i < n$, and $\mathrm{alph}(s_i) \subseteq \mathrm{alphinf}(s), \mathrm{alph}(s'_i) \subseteq \mathrm{alphinf}(s')$ for all $0 < i < n$. If necessary, we use empty factors so that all four products are over the same index set.

We have $\mathrm{alphinf}(s) \times \mathrm{alph}(t) \subseteq I$, hence $st = \prod_{0 \leqslant i < n} s_i t_i$. Since h is weakly alphabetic, we also get $\mathrm{alphinf}(s') \times \mathrm{alph}(t') \subseteq I$ by Remark 21. Therefore $s't' = \prod_{0 \leqslant i < n} s'_i t'_i$ is well-defined, too. Since $h(s_i) = h(s'_i)$ and $h(t_i) = h(t'_i)$, we have $h(s_i t_i) = h(s'_i t'_i)$ for all $0 \leqslant i < n$ and we get $st \sim s't'$. We deduce that $s't' \in [st]$. Since $\approx$ is the transitive closure of $\sim$, a simple induction shows the claim of the lemma. □

We prove Theorem 20 by induction on the monoid M and the alphabet Σ. More precisely, our induction parameter is the pair $(|M|, |\Sigma|)$ and we use the lexicographic order.

The assertion of Theorem 20 is easy if $h(c) = 1_M$ for all $c \in \Sigma$. Indeed, in this case, the set L is a boolean combination of the sets $\{\varepsilon\}, \mathbb{M} \setminus \varepsilon$ and $\mathbb{R} \setminus \mathbb{M}$. Moreover, $\{\varepsilon\} = \mathcal{L}(\bigwedge_{a \in \Sigma} \neg \mathsf{X}_a \top)$ and the set $\mathbb{R} \setminus \mathbb{M}$ of infinite traces is expressed by the formula $\bigvee_{a \in \Sigma} \mathsf{F}^\infty a$ where the macro $\mathsf{F}^\infty a = \mathsf{X}_a \mathsf{G}(\neg a \vee \mathsf{X}_a \top)$ means that there are infinitely many a-labelled vertices above the current one. Note that when $|M| = 1$ or $|\Sigma| = 0$ then we have $h(c) = 1_M$ for all $c \in \Sigma$ and this special case ensures the base of the induction.

We fix in the following some letter $c \in \Sigma$ such that $h(c) \neq 1$. We let $A = \Sigma \setminus \{c\}$ and $\Delta = \mathbb{M}_A(c\mathbb{R} \cap \mathbb{R}^1)$. Recall that $\mathbb{R}_A = \{t \in \mathbb{R} \mid \mathrm{alph}(t) \subseteq A\}$ and $\mathbb{M}_A = \mathbb{R}_A \cap \mathbb{M}$.

Lemma 23. *Let $L \subseteq \mathbb{R}$ be a trace language recognized by the morphism h. Then, $L \setminus \Delta$ is definable by a formula in $\mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$.*

Proof. We have $\mathbb{R} \setminus \Delta = \mathbb{R}_A \cup (\mathbb{R}_A \setminus \mathbb{M}_A)(c\mathbb{R} \cap \mathbb{R}^1)$. Since $L \cap \mathbb{R}_A$ is recognized by the restriction $h{\upharpoonright}_{\mathbb{M}_A}$ of h to $\mathbb{M}_A$ and $|A| < |\Sigma|$ we get by induction a formula ξ_0 for $L \cap \mathbb{R}_A$. Note that, a priori, the induction gives a formula $\xi'_0 \in \mathrm{LocTL}_A[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ such that $L \cap \mathbb{R}_A = \{t \in \mathbb{R}_A \mid \#t \models \xi'_0\}$. Then the formula $\xi_0 = \xi'_0 \wedge \neg\, \mathsf{X}_c \top$ is in $\mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ and $\mathcal{L}(\xi_0) = L \cap \mathbb{R}_A$.

Consider now a trace $t = rcs \in L$ with $r \in \mathbb{R}_A \setminus \mathbb{M}_A$ and $cs \in \mathbb{R}^1$. The language $[r] \cap \mathbb{R}_A$ of traces in $\mathbb{R}_A$ that are h-equivalent to r is recognized by $h{\upharpoonright}_{\mathbb{M}_A}$ hence we get as above a formula $\varphi_{[r]}$ for $[r] \cap \mathbb{R}_A$. We have $\mathrm{alph}(s) \subsetneq \Sigma$ since r is infinite and $\mathrm{alphinf}(r) \times \mathrm{alph}(s) \subseteq I$. Therefore, by induction we find a formula $\psi_{[s]}$ for $\cup_{B \subsetneq \Sigma}[s] \cap \mathbb{R}_B$. By Lemma 5, we may assume that $\psi_{[s]}$ is *iml*. Let $\xi_{[r],[s]} = \overline{\varphi_{[r]}}^A \wedge \mathsf{X}_c\, \psi_{[s]}$ where $\overline{\varphi_{[r]}}^A$ is given by Theorem 11. Note that $\#r = \mu_A(x, \#t)$ where x is the minimal vertex of $\#t$. Hence, $\#t \models \overline{\varphi_{[r]}}^A$ and since x_c is the minimal vertex of cs and the formula $\psi_{[s]}$ is *iml*, we also have $\#t \models \mathsf{X}_c\, \psi_{[s]}$. Therefore, $t \in \mathcal{L}(\xi_{[r],[s]})$.

Let $\varphi = \xi_0 \vee \bigvee_{(u,v) \in W} \xi_{u,v}$ where W is the set of pairs $([r],[s])$ such that $rcs \in L$, $r \in \mathbb{R}_A \setminus \mathbb{M}_A$ and $cs \in \mathbb{R}^1$. We have already shown that $L \setminus \Delta \subseteq \mathcal{L}(\varphi)$.

Conversely, let $t' \in \mathcal{L}(\xi_{[r],[s]})$ where r, s are as above. Define $\#r' = \mu_A(x, \#t')$ where x is now the minimal vertex of $\#t'$. By Theorem 11 we get $r' \in \mathcal{L}(\varphi_{[r]}) = [r] \cap \mathbb{R}_A$. Since $\#t' \models \mathsf{X}_c\, \psi_{[s]}$, x_c exists and with $s' = \Uparrow x_c$ we get $t' = r'cs'$, $cs' \in \mathbb{R}^1$ and $cs' \models \psi_{[s]}$. Since this formula is *iml*, we deduce that $s' \in \mathcal{L}(\psi_{[s]}) \subseteq [s]$. Therefore, $t' = r'cs' \in [r]c[s] \subseteq [rcs] \subseteq L$ by Lemma 22. Therefore, $t' \in L \cap (\mathbb{R}_A \setminus \mathbb{M}_A)(c\mathbb{R} \cap \mathbb{R}^1)$, which concludes the proof. □

We define the notion of *c-factorization* for traces in $\mathbb{R}$. If $t \in \mathbb{R} \setminus \Delta$ then its c-factorization is t itself. The set Δ is a disjoint union of $\Delta_1 = \mathbb{M}_A(c\mathbb{M}_A \cap \mathbb{R}^1)^\omega$ and $\Delta_2 = \mathbb{M}_A(c\mathbb{M}_A \cap \mathbb{R}^1)^*(c(\mathbb{R} \setminus \Delta) \cap \mathbb{R}^1)$. A trace $t \in \Delta_1$ can be written in a unique way as an infinite product (its c-factorization) $t = t_0ct_1ct_2 \cdots$ with $t_0 \in \mathbb{M}_A$ and $ct_i \in c\mathbb{M}_A \cap \mathbb{R}^1$ for all $i > 0$. Similarly, the c-factorization of a trace $t \in \Delta_2$ is the finite product $t = t_0ct_1 \cdots ct_k$ with $t_0 \in \mathbb{M}_A$, $ct_i \in c\mathbb{M}_A \cap \mathbb{R}^1$ for all $0 < i < k$ and $ct_k \in \mathbb{R}^1$ with $t_k \notin \Delta$.

The next step is to replace the c-factorization of t by some sequence over a finite alphabet. For this purpose and for the rest of this section let $T_1 = h(\mathbb{M}_A)$ and $T_2 = \{[s]_h \mid s \in \mathbb{R} \setminus \Delta\}$. We let T be the disjoint union of T_1 and T_2 and we view T as a finite alphabet.

The c-factorization induces a canonical mapping $\sigma : \mathbb{R} \to T^\infty$ as follows. If $t \in \Delta_1$ and its c-factorization is the infinite product $t = t_0ct_1ct_2 \cdots$ then we let $\sigma(t) = h(t_0)h(t_1)h(t_2) \cdots \in T_1^\omega$. If the c-factorization of $t \in (\mathbb{R} \setminus \Delta) \cup \Delta_2$ is the finite product $t = t_0c \cdots ct_k$ $(k \geqslant 0)$ then let $\sigma(t) = h(t_0) \cdots h(t_{k-1})[t_k]_h \in T_1^*T_2$.

Lemma 24. *Let $L \subseteq \mathbb{R}$ be a trace language recognized by the morphism h from $\mathbb{M}$ to M. Then $L = \sigma^{-1}(K)$ for some language K definable in* $\mathrm{LTL}_T[\mathsf{X}, \mathsf{U}]$.

The proof of this lemma uses the induction on the size of the monoid M. The language K will be obtained from languages recognized by a (weakly alphabetic) morphism g from T^* to some monoid M' with $|M'| < |M|$. The monoid M' is obtained with a non-standard construction on monoids. Since this construction might be useful elsewhere, we explain it outside of the proof of Lemma 24. The construction is very similar to a construction of what is known as *local algebra*,[1] see [12,21].[2]

[1] Let A be an associative algebra and $m \in A$. The *local algebra* at m is defined in the literature as mAm with new product $mxm \circ mym = mxmym$.

[2] The reference to [12,21] is due to Benjamin Steinberg.

For a moment let M be any monoid and $m \in M$ an element. Then $mM \cap Mm$ is obviously a sub semigroup, but we emphasize that it is not a monoid, in general. (Note that we do not demand m to be idempotent.) Nevertheless, we can define a new product $\circ$ such that $mM \cap Mm$ becomes a monoid where m is a neutral element: We define $xm \circ my = xmy$ for $xm, my \in mM \cap Mm$. This is well-defined since $xm = x'm$ and $my = my'$ imply $xmy = x'my'$. The operation is associative and $m \circ z = z \circ m = z$. Hence $(mM \cap Mm, \circ, m)$ is indeed a monoid. If M is aperiodic, then $(mM \cap Mm, \circ, m)$ is aperiodic, too. Indeed, if $mx \in Mm$ then, by induction on n, the n-th $\circ$-power of mx is mx^n, hence the result.[3] Moreover, if a finite monoid M is aperiodic with neutral element 1_M and $m \neq 1_M$, then $|mM \cap Mm| < |M|$ since $1_M \notin mM \cap Mm$. Indeed, assume by contradiction that $1_M \in mM \cap Mm$ and write $1_M = mx \in Mm$. Since M is aperiodic, we find $n \geqslant 0$ minimal with $x^n = x^{n+1}$. We have $mx^n = mx^{n+1}$ and since $mx = 1_M$ and n is minimal, we get $n = 0$. But this implies $m = mx = 1_M$, a contradiction.

Proof of Lemma 24. Let M be again the finite aperiodic monoid we fixed above together with the morphism h. Then $h(c) \neq 1$ and the monoid $M' = (h(c)M \cap Mh(c), \circ, h(c))$ has a smaller size than M. Let us define a morphism $g : T^* \to M'$ as follows. For $m = h(s) \in T_1$ we define $g(m) = h(c)mh(c) = h(csc)$. For $m \in T_2$ we let $g(m) = h(c)$, which is the neutral element in M'.

Let $K_0 = \{[s]_h \mid s \in L \setminus \Delta\}$. We claim that $L \setminus \Delta = \sigma^{-1}(K_0)$. One inclusion is clear. Conversely, let $t \in \sigma^{-1}(K_0)$. There exists $s \in L \setminus \Delta$ such that $\sigma(t) = [s]_h$. By definition of σ, this implies $t \notin \Delta$ and $\sigma(t) = [t]_h$. Since $s \in L$ and L is recognized by h, we get $t \in L$ as desired.

For $n \in T_1$ and $m \in T_2$, let $K_{n,m} = nT_1^*m \cap n[n^{-1}\sigma(L) \cap T_1^*m]_g$ and let $K_2 = \bigcup_{n \in T_1, m \in T_2} K_{n,m}$. We claim that $L \cap \Delta_2 = \sigma^{-1}(K_2)$. Let first $t \in L \cap \Delta_2$ and write $t = t_0ct_1 \cdots ct_k$ its c-factorization. With $n = h(t_0)$ and $m = [t_k]_h$ we get $\sigma(t) \in K_{n,m}$. Conversely, let $t \in \sigma^{-1}(K_{n,m})$ with $n \in T_1$ and $m \in T_2$. We have $t \in \Delta_2$ and its c-factorization is $t = t_0ct_1 \cdots ct_k$ with $h(t_0) = n$ and $[t_k]_h = m$ $(k > 0)$. Moreover, $x = h(t_1) \cdots h(t_{k-1})[t_k]_h \in [n^{-1}\sigma(L) \cap T_1^*m]_g$ hence we find $y \in T_1^*m$ with $g(x) = g(y)$ and $ny \in \sigma(L)$. Let $s \in L$ be such that $\sigma(s) = ny \in nT_1^*m$. Then $s \in \Delta_2$ and its c-factorization is $s = s_0cs_1 \cdots cs_\ell$ with $h(s_0) = n$ and $[s_\ell]_h = m$ $(\ell > 0)$. By definition of g, we get $h(ct_1c \cdots ct_{k-1}c) = g(x) = g(y) = h(cs_1c \cdots cs_{\ell-1}c)$ and we deduce that $t \approx_h s$. Since $s \in L$ and L is recognized by h, we get $t \in L$ as desired.

For $n \in T_1$, let now $K_{n,\omega} = nT_1^\omega \cap n[n^{-1}\sigma(L) \cap T_1^\omega]_g$ and let $K_1 = \bigcup_{n \in T_1} K_{n,\omega}$. As above, we will show that $L \cap \Delta_1 = \sigma^{-1}(K_1)$. So let $t \in L \cap \Delta_1$ and consider its c-factorization $t = t_0ct_1ct_2 \cdots$. With $n = h(t_0)$, we get $\sigma(t) \in K_{n,\omega}$. To prove the converse inclusion we need some auxiliary results.

First, if $x \sim_g y \sim_g z$ with $x \in T^\omega$ and $|y|_{T_1} < \omega$ then $x \sim_g z$. Indeed, in this case, we find factorizations $x = x_0x_1x_2 \cdots$ and $y = y_0y_1y_2 \cdots$ with $x_i \in T^+$, $y_0 \in T^+$ and $y_i \in T_2^+$ for $i > 0$ such that $g(x_i) = g(y_i)$ for all $i \geqslant 0$. Similarly, we find factorizations $z = z_0z_1z_2 \cdots$ and $y = y'_0y'_1y'_2 \cdots$ with $z_i \in T^+$, $y'_0 \in T^+$ and $y'_i \in T_2^+$ for $i > 0$ such that $g(z_i) = g(y'_i)$ for all $i \geqslant 0$. Then, we have $g(x_i) = g(y_i) = h(c) = g(y'_i) = g(z_i)$ for all $i > 0$ and $g(x_0) = g(y_0) = g(y'_0) = g(z_0)$ since y_0 and y'_0 contain all letters of y from T_1 and g maps all letters from T_2 to the neutral element of M'.

Second, if $x \sim_g y \sim_g z$ with $|y|_{T_1} = \omega$ then $x \sim_g y' \sim_g z$ for some $y' \in T_1^\omega$. Indeed, in this case, we find factorizations $x = x_0x_1x_2 \cdots$ and $y = y_0y_1y_2 \cdots$ with $x_i \in T^+$, and $y_i \in T^*T_1T^*$ such that $g(x_i) =$

[3] As Daniel Kirsten pointed out $(mM \cap Mm, \circ, m)$ is in fact a divisor of M: Let $M^{(m)} = \{x \in M \mid xm \in mM\}$. Then $M^{(m)}$ is a submonoid of M, and the mapping $f(x) = xm$ is a surjective morphism from $M^{(m)}$ onto $(mM \cap Mm, \circ, m)$.

$g(y_i)$ for all $i \geqslant 0$. Let y'_i be the projection of y_i to the subalphabet T_1 and let $y = y'_0 y'_1 y'_2 \cdots \in T_1^\omega$. We have $g(y_i) = g(y'_i)$, hence $x \sim_g y'$. Similarly, we get $y' \sim_g z$.

Third, if $\sigma(t) \sim_g \sigma(s)$ with $t, s \in \Delta_1$ then $ct \approx_h cs$. Indeed, since $t, s \in \Delta_1$, the c-factorizations of t and s are of the form $t_1 c t_2 \cdots$ and $s_1 c s_2 \cdots$. Using $\sigma(t) \sim_g \sigma(s)$, we find new factorizations $t = t'_1 c t'_2 \cdots$ and $s = s'_1 c s'_2 \cdots$ with $t'_i, s'_i \in \mathbb{M}_A(c\mathbb{M}_A \cap \mathbb{R}^1)^+$ and $h(ct'_i c) = h(cs'_i c)$ for all $i > 0$. We deduce

$$\begin{aligned} ct = (ct'_1 c)t'_2(ct'_3 c)t'_4 \cdots \sim_h\ & (cs'_1 c)t'_2(cs'_3 c)t'_4 \cdots = \\ & cs'_1(ct'_2 c)s'_3(ct'_4 c) \cdots \sim_h cs'_1(cs'_2 c)s'_3(cs'_4 c) \cdots = cs. \end{aligned}$$

We come back to the proof of $\sigma^{-1}(K_{n,\omega}) \subseteq L \cap \Delta_1$. So let $t \in \sigma^{-1}(K_{n,\omega})$. We have $t \in \Delta_1$ and $\sigma(t) = nx \in nT_1^\omega$ with $x \in [n^{-1}\sigma(L) \cap T_1^\omega]_g$. Let $y \in T_1^\omega$ be such that $x \approx_g y$ and $ny \in \sigma(L)$. Let $s \in L$ with $\sigma(s) = ny$. We may write $t = t_0 ct'$ and $s = s_0 cs'$ with $t_0, s_0 \in \mathbb{M}_A$, $h(t_0) = n = h(s_0)$, $ct', cs' \in \mathbb{R}^1$, $x = \sigma(t')$ and $y = \sigma(s')$. Since $x \approx_g y$, using the first two auxiliary results above and the fact that the mapping $\sigma : \Delta_1 \to T_1^\omega$ is surjective, we get $\sigma(t') \sim_g \sigma(r_1) \sim_g \cdots \sim_g \sigma(r_k) \sim_g \sigma(s')$ for some $r_1, \ldots, r_k \in \Delta_1$. From the third auxiliary result, we get $ct' \approx_h cs'$. Hence, using $h(t_0) = h(s_0)$, we obtain $t = t_0 ct' \approx_h s_0 cs' = s$. Since $s \in L$ and L is recognized by h, we get $t \in L$ as desired.

Finally, let $K = K_0 \cup K_1 \cup K_2$. We have already seen that $L = \sigma^{-1}(K)$. It remains to show that K is definable in $\mathrm{LTL}_T[\mathsf{X}, \mathsf{U}]$. Let $N \subseteq T^\infty$, then, by definition, the language $[N]_g$ is recognized by g which is a weakly alphabetic morphism to the aperiodic monoid M' with $|M'| < |M|$. By induction on the size of the monoid, we deduce that all languages of the form $[N]_g$ are definable in $\mathrm{LocTL}_T[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ and hence in $\mathrm{LTL}_T[\mathsf{X}, \mathsf{U}]$ by Proposition 2 since for words, EX is the usual X modality.[4] Now, if a language $N \subseteq T^\infty$ is defined by $f \in \mathrm{LTL}_T[\mathsf{X}, \mathsf{U}]$ and $n \in T$ then the language nN is defined by $n \wedge \mathsf{X} f$. Moreover, K_0, $nT_1^* m$ and nT_1^ω are obviously definable in $\mathrm{LTL}_T(\mathsf{X}, \mathsf{U})$. Therefore, K is definable in $\mathrm{LTL}_T[\mathsf{X}, \mathsf{U}]$. □

The next lemma yields the basic transformation from an LTL formula over words to a formula in local temporal logic over traces.

Lemma 25. *For each formula $f \in \mathrm{LTL}_T[\mathsf{X}, \mathsf{U}]$ there exists a formula $\widetilde{f} \in \mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ such that for all $t \in \mathbb{R}$ we have $\sigma(t) \models f$ if and only if $\#t \models \widetilde{f}$.*

Proof. Clearly, we have $\widetilde{\bot} = \bot$, $\widetilde{\neg f} = \neg \widetilde{f}$ and $f_1 \widetilde{\vee} f_2 = \widetilde{f_1} \vee \widetilde{f_2}$.

Now, we consider the case $f = m \in T_1$. For $t \in \mathbb{R}$ we have $\sigma(t) \models m$ if and only if $t = rcs$ with $r \in h^{-1}(m) \cap \mathbb{M}_A$ and $cs \in \mathbb{R}^1$. Clearly, $h^{-1}(m) \cap \mathbb{M}_A$ is recognized by $h\restriction_{\mathbb{M}_A}$ and as in the proof of Lemma 23, we get by induction on the size of the alphabet a formula $\varphi_m \in \mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ such that $h^{-1}(m) \cap \mathbb{M}_A = \mathcal{L}(\varphi_m)$. Using Theorem 11 we obtain: $\widetilde{m} = \overline{\varphi_m}^A \wedge \mathsf{X}_c \top$. Indeed, assume that $\#t \models \widetilde{m}$ for some $t \in \Delta$. Let $\#r = \mu_A(\#t, x)$ where x is the minimal vertex of $\#t$. Since $\#t \models \mathsf{X}_c \top$ we have $t = rcs$ for some s with $cs \in \mathbb{R}^1$. Now, by Theorem 11 we get $\#r \models \varphi_m$. Hence, $r \in h^{-1}(m) \cap \mathbb{M}_A$ and $\sigma(t) \models m$. The converse can be shown similarly.

Next, assume that $f = m = [s]_h \in T_2$. We have $\sigma(t) \models m$ if and only if $t \in [s]_h \setminus \Delta$. The result follows by Lemma 23.

[4] The statement that an aperiodic language K over words in T^∞ is definable in $\mathrm{LTL}_T[\mathsf{XU}]$ is also a consequence of classical papers [30,18,23,34,13,27,28,3]. Therefore it is of course a well-known result but we do not need it since we get it for free by induction on the monoid size.

Finally, it is well-known that, for words, the logic $\mathrm{LTL}_T[\mathsf{X},\mathsf{U}]$ is equivalent to $\mathrm{LTL}_T[\mathsf{XU}]$ where $f_1\,\mathsf{XU}\,f_2 = \mathsf{X}(f_1\,\mathsf{U}\,f_2)$. Hence, it remains to deal with the modality XU. For this we use the fact that $\Delta = \mathcal{L}(\delta)$ where

$$\delta = \neg\,\mathsf{X}_c \top \vee \bigwedge_{a\in A} (\mathsf{F}^\infty a \Longleftrightarrow \mathsf{X}_c\,\mathsf{F}^\infty a).$$

Note that $\mathsf{F}^\infty a = \mathsf{X}_a\,\mathsf{G}(\neg a \vee \mathsf{X}_a \top)$ is an *iml* formula, hence δ is *iml*, too. Now, we claim that $f_1\,\widetilde{\mathsf{XU}}\,f_2 = \delta \wedge \mathsf{X}_c((\delta \wedge \widetilde{f_1})\,\mathsf{U}_c\,\widetilde{f_2})$, where we assume using Lemma 5 that $\widetilde{f_1}$ and $\widetilde{f_2}$ are *iml*. To see this, assume first that $\#t \models f_1\,\widetilde{\mathsf{XU}}\,f_2$ and write $t = t_0ct_1\cdots ct_j$ with $t_0 \in \mathbb{R}_A$, $ct_i \in (c\mathbb{R}_A \cap \mathbb{R}^1)$ for $0 < i < j$, $ct_j \in \mathbb{R}^1$, $ct_i\cdots ct_j \models \delta \wedge \widetilde{f_1}$ for $0 < i < j$ and $ct_j \models \widetilde{f_2}$. Since $t \models \delta$ and $ct_i \cdots ct_j \models \delta$ for $0 < i < j$, we deduce that $t_i \in \mathbb{M}_A$ for $0 \leqslant i < j$. Hence, $\sigma(t) = h(t_0)\cdots h(t_{j-1})\sigma(t_j)$. The formula $\widetilde{f_2}$ is *iml*, hence $\#t_j \models \widetilde{f_2}$ and by induction we obtain $\sigma(t_j) \models f_2$. Similarly, since $\widetilde{f_1}$ is *iml*, we get $\sigma(t_ic\cdots ct_j) = h(t_i)\cdots h(t_{j-1})\sigma(t_j) \models f_1$ for $0 < i < j$. Therefore, $\sigma(t) \models f_1\,\mathsf{XU}\,f_2$ as required. The proof for the converse is similar. □

Theorem 20 is a direct consequence of Lemmas 24 and 25. By Proposition 2 and Theorem 17 we obtain:

Corollary 26. *Let $L \subseteq \mathbb{R}(\Sigma, D)$ be a real trace language. The following assertions are equivalent*:

(1) *The language L is expressible in* $\mathrm{FO}_\Sigma(<)$.
(2) *We have $L = \mathcal{L}_\Sigma(\varphi)$ for some* $\varphi \in \mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$.
(3) *We have $L = \mathcal{L}_\Sigma(\varphi)$ for some* $\varphi \in \mathrm{LocTL}_\Sigma[\mathsf{X}_i, \mathsf{U}_i]$.
(4) *We have $L = \mathcal{L}_\Sigma(\varphi)$ for some* $\varphi \in \mathrm{LocTL}_\Sigma[\mathsf{EX}, \mathsf{U}]$.
(5) *We have $L = \mathcal{L}_\Sigma(\varphi)$ for some* $\varphi \in \mathrm{LocTL}_\Sigma[\mathsf{SU}]$.

We obtain also easily the same equivalence for trace languages in $\mathbb{R}^1$.

Corollary 27. *Let $L \subseteq \mathbb{R}^1$ be a language of real traces having a unique minimal vertex. The following assertions are equivalent*:

(1) *The language L is expressible in* $\mathrm{FO}_\Sigma(<)$.
(2) *We have $L = \mathcal{L}^1(\varphi)$ for some* $\varphi \in \mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$.
(3) *We have $L = \mathcal{L}^1(\varphi)$ for some* $\varphi \in \mathrm{LocTL}_\Sigma[\mathsf{X}_i, \mathsf{U}_i]$.
(4) *We have $L = \mathcal{L}^1(\varphi)$ for some* $\varphi \in \mathrm{LocTL}_\Sigma[\mathsf{EX}, \mathsf{U}]$.
(5) *We have $L = \mathcal{L}^1(\varphi)$ for some* $\varphi \in \mathrm{LocTL}_\Sigma[\mathsf{SU}]$.

Proof. In view of Proposition 2 we only need to show 1 implies 2. So let $L \subseteq \mathbb{R}^1$ be expressible in $\mathrm{FO}_\Sigma(<)$. We have $L = \bigcup_{c\in\Sigma} c \cdot (c^{-1}L)$ and each language $c^{-1}L = \{t \in \mathbb{R} \mid ct \in L\}$ is also expressible in $\mathrm{FO}_\Sigma(<)$. By Theorem 17 we find a formula $\varphi_c \in \mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_a, \mathsf{U}_a]$ such that $c^{-1}L = \mathcal{L}(\varphi_c)$ and we may assume that φ_c is *iml* by Lemma 5. We get $L = \mathcal{L}^1(\varphi)$ with $\varphi = \bigvee_{c\in\Sigma} c \wedge \varphi_c$. Indeed, let $ct \in L$. We have $t \in c^{-1}L$ hence $\#t \models \varphi_c$. We get $ct \models c \wedge \varphi_c$ since φ_c is *iml*. Conversely, assume that $s \in \mathcal{L}^1(c \wedge \varphi_c)$ for some $c \in \Sigma$. Then, $s = ct$ and $\#t \models \varphi_c$ since this formula is *iml*. Therefore, $t \in c^{-1}L$ and $s \in c \cdot (c^{-1}L) \subseteq L$. □

7. Concluding remarks

Since the result of this paper has been obtained in fall 2003, we have continued the research in the following directions. In [8] we proved that our result in [4] on the expressive completeness of the global temporal logic can be derived quite easily from the results of the present paper. We have also started, but did not finish yet, an investigation on *local safety* properties. This is quite subtle and indicates that this concept is related to the notion of *coherent closure* rather than to a pure topological concept (Scott closure), as over words or in a global semantics [5].

Many other problems remain open. As we have seen the 3-variable fragment of FO($<$) has the same expressive power as the full first-order theory FO($<$). The 2-variable fragment of FO($<$) is weaker. Over finite words its expressive power is well-understood. The 2-variable fragment corresponds to $\mathrm{LocTL}_\Sigma[\mathsf{XF}, \mathsf{YP}]$ which is equal to $\mathrm{LocTL}_\Sigma[\mathsf{X}_a, \mathsf{Y}_a]$, and it can be algebraically characterized by the variety **DA**, [31]. Here XF means *Next-Future* and YP means *Yesterday-Past*. Hence $t, x \models \mathsf{XF}\,\varphi$ if $t, y \models \varphi$ for some node y strictly above x (i.e., $x < y$). The operator YP is dual.

In the presence of independence the situation is more complicated. With two variables we can express that a trace contains 2 parallel nodes. This leads out of the variety **DA**. In his Ph.D. thesis [20], Kufleitner showed that for finite traces we still have the correspondences between $\mathrm{LocTL}_\Sigma[\mathsf{XF}, \mathsf{YP}]$, $\mathrm{LocTL}_\Sigma[\mathsf{X}_a, \mathsf{Y}_a]$, and **DA**, but these fragments are weaker than the 2-variable fragment of FO($<$). It is an interesting open problem whether the 2-variable fragment of FO($<$) is decidable, in general. Indeed, compared to the rich theory of regular word languages very little is known for recognizable trace languages.

Acknowledgments

We thank the anonymous referees for reading the manuscript and for asking whether the logics $\mathrm{LocTL}_\Sigma[X_i, \mathsf{U}_i]$ and $\mathrm{LocTL}_\Sigma[\mathsf{X}_a, \mathsf{U}_a]$ were expressively complete. This leads us to strengthen Proposition 2 and to add Remark 4. In the submitted version, we only proved that the logic $\mathrm{LocTL}_\Sigma[(\mathsf{X}_a \leqslant \mathsf{X}_b), \mathsf{X}_i, \mathsf{U}_i]$ is expressively complete since we did not try to express the constants $(\mathsf{X}_a \leqslant \mathsf{X}_b)$ in $\mathrm{LocTL}_\Sigma[\mathsf{X}_i, \mathsf{U}_i]$.

References

[1] B. Adsul, M. Sohoni, Complete and tractable local linear time temporal logics over traces, in: Proc. ICALP'02 number 2380 in LNCS, Springer, New York, 2002, pp. 926–937.

[2] R. Alur, D. Peled, W. Penczek, Model-checking of causality properties, in: Proc. LICS'95, IEEE Computer Society Press, Silver Spring, MD, 1995, pp. 90–100.

[3] J. Cohen, D. Perrin, J.-E. Pin, On the expressive power of temporal logic, J. Comput. Syst. Sci. 46 (1993) 271–295.

[4] V. Diekert, P. Gastin, LTL is expressively complete for Mazurkiewicz traces, J. Comput. Syst. Sci. 64 (2002) 396–418, A preliminary version appeared at ICALP'00, LNCS 1853, Springer, New York, pp. 211–222.

[5] V. Diekert, P. Gastin, Safety and liveness properties for real traces and a direct translation from LTL to monoids, in: Formal and Natural Computing—Essays Dedicated to Grzegorz Rozenberg, number 2300 in LNCS, Springer, New York, 2002, pp. 26–38.

[6] V. Diekert, P. Gastin, Local temporal logic is expressively complete for cograph dependence alphabets, Informat. Comput. 195 (2004) 30–52, A preliminary version appeared at LPAR'01, LNAI 2250, Springer, New York, pp. 55–69.

[7] V. Diekert, P. Gastin, Pure future local temporal logics are expressively complete for Mazurkiewicz traces, in: M. Farach-Colton (Ed.), Proc. LATIN'04, number 2976 in LNCS, Springer, New York, 2004, pp. 232–241.

[8] V. Diekert, P. Gastin, From local to global temporal logics over Mazurkiewicz traces, Theor. Comput. Sci. 356 (1-2) (2006) 126–135, In honour of Professor Christian Choffrut on the occasion of his 60th birthday.

[9] in: V. Diekert, G. Rozenberg (Eds.), The Book of Traces, World Scientific, Singapore, 1995.

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

[11] W. Ebinger, A. Muscholl, Logical definability on infinite traces, Theor. Comput. Sci. 154 (1996) 67–84.

[12] A. Ferandez Lopez, M.I. Tocon Barroso, The local algebras of an associative algebra and their applications, in: J.C. Misra (Ed.), Applicable Mathematics in the Golden Age, Narosa, 2002, pp. 254–275.

[13] D. Gabbay, A. Pnueli, S. Shelah, J. Stavi, On the temporal analysis of fairness, in: Proc. PoPL'80, Las Vegas, Nev., 1980, pp. 163–173.

[14] P. Gastin, D. Kuske, Satisfiability and model checking for MSO-definable temporal logics are in PSPACE, in: Proc. CONCUR'03, number 2761 in LNCS, Springer, New York, 2003, pp. 222–236.

[15] P. Gastin, M. Mukund, An elementary expressively complete temporal logic for Mazurkiewicz traces, in: Proc. ICALP'02, number 2380 in LNCS, Springer, New York, 2002, pp. 938–949.

[16] P. Gastin, M. Mukund, K. Narayan Kumar, Local LTL with past constants is expressively complete for Mazurkiewicz traces, in: Proc. MFCS'03, number 2747 in LNCS, Springer, New York, 2003, pp. 429–438.

[17] P. Gastin, A. Petit, Infinite traces, in: V. Diekert, G. Rozenberg (Eds.), The Book of Traces, World Scientific, Singapore, 1995, pp. 393–486, chapter 11.

[18] J.A.W. Kamp, Tense Logic and the Theory of Linear Order. PhD thesis, University of California, Los Angeles, California, 1968.

[19] R.M. Keller, Parallel program schemata and maximal parallelism I. Fundamental results, J. Assoc. Comput. Mach. 20 (3) (1973) 514–537.

[20] M. Kufleitner, Logical Fragments for Mazurkiewicz Traces: Expressive Power and Algebraic Characterizations. Ph.D. thesis, Institut für Formale Methoden der Informatik, Universität Stuttgart, 2006.

[21] K. Mayberg, Lectures on algebras and triple systems, Technical report, University of Virginia, Charlottesville, 1972.

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

[23] R. McNaughton, S. Papert, Counter-Free Automata, MIT Press, Cambridge, MA, 1971.

[24] M. Mukund, P.S. Thiagarajan, Linear time temporal logics over Mazurkiewicz traces, in: Proc. MFCS'96, number 1113 in LNCS, Springer, New York, 1996, pp. 62–92.

[25] P. Niebert, A ν-calculus with local views for sequential agents, in: Proc. MFCS'95, number 969 in LNCS, Springer, New York, 1995, pp. 563–573.

[26] W. Penczek, Temporal logics for trace systems: On automated verification, Intl. J. Foundation Comput. Sci. 4 (1993) 31–67.

[27] D. Perrin, Recent results on automata and infinite words, in: Proc. MFCS'84, number 176 in LNCS, Springer, New York, 1984, pp. 134–148.

[28] D. Perrin, J.-E. Pin, First order logic and star-free sets, J. Comput. Syst. Sci. 32 (1986) 393–406.

[29] R. Ramanujam, Locally linear time temporal logic, in: Proc. LICS'96, 1996, pp. 118–128.

[30] M.-P. Schützenberger, On finite monoids having only trivial subgroups, Inform. Control 8 (1965) 190–194.

[31] P. Tesson, D. Thérien, Diamonds are Forever: The Variety DA, in: Semigroups, Algorithms, Automata and Languages, Coimbra (Portugal) 2001, World Scientific, Singapore, 2002, pp. 475–500.

[32] P.S. Thiagarajan, A trace based extension of linear time temporal logic, in: Proc. of LICS'94, IEEE Computer Society Press, Silver Spring, MD, 1994, pp. 438–447.

[33] P.S. Thiagarajan, I. Walukiewicz, An expressively complete linear time temporal logic for Mazurkiewicz traces, in: Proc. LICS'97, 1997, pp. 183–194.

[34] W. Thomas, Star-free regular sets of ω-sequences, Inform. Control 42 (1979) 148–156.

[35] I. Walukiewicz, Difficult configurations—on the complexity of LTrL, in: Proc. ICALP'98, number 1443 in LNCS, Springer, New York, 1998, pp. 140–151.

[36] I. Walukiewicz, Private communication, 2001.

[37] Th. Wilke, Classifying discrete temporal properties, in: Proc. STACS'99, number 1563 in LNCS, Springer, New York, 1999, pp. 32–46.