Local Temporal Logic is Expressively Complete for Cograph Dependence Alphabets ⋆

Volker Diekert[1] and Paul Gastin[2]

[1] Inst. für Informatik, Universität Stuttgart, Breitwiesenstr. 20-22, D-70565 Stuttgart
diekert@informatik.uni-stuttgart.de
[2] LIAFA, Université Paris 7, 2, place Jussieu, F-75251 Paris Cedex 05
Paul.Gastin@liafa.jussieu.fr

Abstract. Recently, local logics for Mazurkiewicz traces are of increasing interest. This is mainly due to the fact that the satisfiability problem has the same complexity as in the word case. If we focus on a purely local interpretation of formulae at vertices (or events) of a trace, then the satisfiability problem of linear temporal logics over traces turns out to be PSPACE–complete. But now the difficult problem is to obtain expressive completeness results with respect to first order logic.
The main result of the paper shows such an expressive completeness result, if the underlying dependence alphabet is a cograph, i.e., if all traces are series parallel graphs. Moreover, we show that this is the best we can expect in our setting: If the dependence alphabet is not a cograph, then we cannot express all first order properties.
Keywords Temporal logics, Mazurkiewicz traces, concurrency

1 Introduction

Trace theory, initiated in computer science by Mazurkiewicz [12] is one of the most popular settings to study concurrency. The behavior of a concurrent process is not represented by a string, but more accurately by some labelled partial order.

A suitable way for a formal specification of concurrent systems is given by temporal logic formulae which in turn 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 [1, 13–16, 18]. In [17] it was shown that the basic (global) linear temporal logic with future tense operators and with past tense constants is expressively complete with respect to the first order theory of finite and infinite traces (*real traces*). In [3] we have obtained the same result but without any past tense modalities by quite different proof techniques (which will be used here again). This positive result has solved a long standing open question [6, 17]. The price of this logic is an extremely difficult satisfiability problem, it has been shown to be non-elementary by Walukiewicz [20]. The main

⋆ Partial support of EC-FET project IST-1999-29082 (ADVANCE), CEFIPRA-IFCPAR Project 2102-1 (ACSMV) and PROCOPE project 00269SD (MoVe) is gratefully acknowledged.

reason for this difficulty is the global interpretation of a formula which makes it necessary to speak about configurations, i.e., we give an interpretation of a formula for a trace with respect to some finite prefix – and the prefix structure of a trace is much more complicated than in the case of linear orders (words). If we give a local interpretation such that each formula can be evaluated at a single vertex (or event), then we obtain logics where the satisfiability problem is still in PSPACE. This is in particular the case for the logic TLC, which has been introduced by Alur et al. in [1]. The logic has been extended and studied in detail by Henriksen in [8, 9]. The logic TLC uses an existential version of the until–operator, which is not expressible in first order, in general (Section 5). On the other hand, it seems also too weak to express all first order properties for all dependence alphabets.

In our paper, we shall use a universal version of the until–operator, which, by its very definition, is first order definable. The main result of our paper shows that we obtain a local logic which is expressively complete, if the underlying dependence graph is a cograph, i.e., every trace is a series parallel graph (or N–free). This result is robust, the same holds for TLC or other variants how to define a semantics to the until– operator. Moreover, we show that cographs are in some sense the limit where we can expect such a positive result. As long as we use no past tense modalities we cannot specify all first order properties by our logic, whether or not the until is existential or universal or whether we have both options. Our main theorems (Thms. 2, 3) are therefore *if-and-only-if* statements.

In the final section we will see that a universal until does not change complexity issues very much. The satisfiability problem of TLC augmented by this operator can still be solved in PSPACE.

For lack of space, we cannot give full proofs in this extended abstract.

2 Preliminaries

We briefly recall some notions concerning Mazurkiewicz traces. For the background we refer to [4]. A *dependence alphabet* is a pair (Σ, D) where the alphabet Σ is a finite set and the dependence relation $D \subseteq \Sigma \times \Sigma$ is reflexive and symmetric. The *independence relation* I is the complement of D. For $A \subseteq \Sigma$, we denote $I(A) = \{b \in \Sigma \mid (a, b) \in I \text{ for all } a \in A\}$ the set of letters independent from A and we let $D(A) = \Sigma \setminus I(A)$ be the set of letters depending on (some action in) A.

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

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

We define the concatenation of two traces $t_1 = [V_1, \leq_1, \lambda_1] \in \mathbb{R}$ and $t_2 = [V_2, \leq_2, \lambda_2] \in \mathbb{R}$ verifying $\mathrm{alphinf}(t_1) \times \mathrm{alph}(t_2) \subseteq I$ by $t_1 \cdot t_2 = [V, \leq, \lambda]$ 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 $\leq$ is the transitive closure of the relation $\leq_1 \cup \leq_2 \cup (V_1 \times V_2 \cap \lambda^{-1}(D))$. The set of finite traces becomes a monoid which is denoted by $\mathbb{M}(\Sigma, D)$ or simply by $\mathbb{M}$. The empty trace $1 = (\emptyset, \emptyset, \emptyset)$ is the unit element. Also, if $A \subseteq \Sigma$, we let $\mathbb{R}_A = \{x \in \mathbb{R} \mid \mathrm{alph}(x) \subseteq A\}$ and $\mathbb{M}_A = \mathbb{M} \cap \mathbb{R}_A$. We also define $\mathbb{M}^+ = \mathbb{M} \setminus \{1\}$ and $\mathbb{M}_A^+ = \mathbb{M}^+ \cap \mathbb{M}_A$.

Our main results concern dependence alphabets which are cographs. According to standard graph theoretical notions, a dependence alphabet is called a *cograph*, if it belongs to the smallest class of graphs which contains singletons and which is closed under the operations of disjoint union and complementation. Clearly, (Σ, D) is a cograph if and only if the *independence alphabet* (Σ, I) is a cograph. It is well-known that an undirected graph is a cograph if and only if it does not contain any P_4 (a line of 4 vertices) as an induced subgraph [2]. It turns out that (Σ, D) is a cograph if and only if all $t \in \mathbb{M}(\Sigma, D)$ are *series parallel graphs*, i.e., we can build up the trace starting with letters by taking serial and parallel products. In particular, every such trace is N–free, i.e., whenever there are four vertices a, b, c, d with $a < b$, $c < b$, and $c < d$, then there is at least one more ordering between them.

We shall use the algebraic notion for recognizability: Let $h : \mathbb{M} \to S$ be a morphism to some finite monoid S. For $x, y \in \mathbb{R}$, we say that x and y are h-similar, denoted by $x \sim_h y$ if either $x, y \in \mathbb{M}$ and $h(x) = h(y)$ or x and y have infinite factorizations in non-empty finite traces $x = x_1 x_2 \cdots$, $y = y_1 y_2 \cdots$ with $x_i, y_i \in \mathbb{M}^+$ and $h(x_i) = h(y_i)$ for all i. We denote by $\approx_h$ the transitive closure of $\sim_h$ which is therefore an equivalence relation. Since S is finite, this equivalence relation is of finite index with at most $|S|^2 + |S|$ equivalence classes. A real trace language $L \subseteq \mathbb{R}$ is *recognized* by h if it is saturated by $\sim_h$, i.e., $x \in L$ implies $[x]_{\approx_h} \subseteq L$ for all $x \in \mathbb{R}$.

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

A finite monoid S is *aperiodic*, if there is some $n \geq 0$ such that $s^n = s^{n+1}$ for all $s \in S$. A real trace language $L \subseteq \mathbb{R}$ is *aperiodic* if it is recognized by some morphism to some finite and aperiodic monoid. We denote by $\mathrm{AP}(\Sigma, D)$ or simply by AP the set of aperiodic languages $L \subseteq \mathbb{R}(\Sigma, D)$. If $A \subseteq \Sigma$, we use the notation AP_A for the aperiodic languages over $\mathbb{R}_A$.

In the following we shall use the well-known equivalence between first-order definability and aperiodic languages. The first order theory of traces is given by the syntax of $\mathrm{FO}_\Sigma(<)$:

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

where $a \in \Sigma$ and $x, y \in \mathrm{Var}$ are first order variables. Given a trace $t = [V, \leq, \lambda]$ and a valuation σ of the free variables into the vertices, the semantics is obtained by interpreting the predicate $P_a(x)$ by $\lambda(\sigma(x)) = a$ and the relation $<$ as the strict partial order relation of the trace t. Then we can say whether or not $t, \sigma \models \varphi$. If φ is a sentence, i.e., a closed formula, then we define the language $\mathcal{L}(\varphi) = \{t \in \mathbb{R} \mid t \models \varphi\}$. We say that a trace language $L \subseteq \mathbb{R}$ is expressible in $\mathrm{FO}_\Sigma(<)$ if there exists some sentence $\varphi \in \mathrm{FO}_\Sigma(<)$ such that $L = \mathcal{L}(\varphi)$. We denote by $\mathrm{FO}_{(\Sigma,D)}(<)$ the set of real trace languages $L \subseteq \mathbb{R}(\Sigma, D)$ such that for some sentence $\varphi \in \mathrm{FO}_\Sigma(<)$ we have $L = \mathcal{L}(\varphi)$

Theorem 1 ([6, 5]). *A language $L \subseteq \mathbb{R}(\Sigma, D)$ is expressible in $\mathrm{FO}_\Sigma(<)$ if and only if it is aperiodic, i.e., $\mathrm{FO}_{(\Sigma,D)}(<) = \mathrm{AP}(\Sigma, D)$.*

We say that a first order formula is in $\mathrm{FO}^n_\Sigma(<)$ if it uses at most n first order variables (it may use each variable several times).

3 Local temporal logics for traces

In this section, we introduce the *local temporal logic* over traces and its semantics. We restrict ourselves to future modalities (partly for lack of space), but of course, dual past tense modalities can also be defined and whenever we state that a fragment is in $\mathrm{FO}_\Sigma(<)$, then the fragment augmented by the dual past tenses will have the same property. In the next sections we specialize the logics by considering various subsets of the modalities. We say that a temporal logic over traces is *local* if it is evaluated at the vertices of the trace as for first order formulae. This is in contrast with global temporal logic formulae that are evaluated at global configurations of the trace, i.e., at finite prefixes of the trace.

For global formulae, we say that a trace is a model of a formula if it satisfies the formula at the empty configuration. There is no such canonical way to interpret a formula at some trace without fixing some vertex since there is no canonical vertex in the trace where to start the evaluation of the formula. Natural vertices are the minimal ones but a trace may have several minimal vertices. We have chosen to introduce initial formulae to address this problem. There are other possibilities, like adding a unique minimal dummy, but then the logic becomes more expressive and we are mainly interested in expressive completeness (with respect to first-order) for a weak fragment of our logic.

We start with the definition of (*internal*) formulae that are evaluated at vertices. The syntax of $\mathrm{LocTL}^{\mathsf{i}}_\Sigma(\mathsf{EX}, \mathsf{U}, \mathsf{EU})$ is given by

$$\varphi ::= \bot \mid a \in \Sigma \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \varphi\,\mathsf{EU}\,\varphi.$$

The symbol $\bot$ means *false*, $\mathsf{EX}\,\varphi$ claims that φ holds for some immediate successor of the current vertex; $\varphi\,\mathsf{U}\,\psi$ is a *universal* until claiming that ψ holds for some vertex above the current one and that φ holds for all vertices in between; on the contrary, $\varphi\,\mathsf{EU}\,\psi$ is an *existential* until claiming the existence of some path starting at the current vertex in the Hasse diagram of the trace where φ holds until ψ does. Formally, the semantics is inductively given as follows. Let $t = [V, \leq, \lambda] \in \mathbb{R}$ and let $x \in V$ (we also write $x \in t$).

$$
\begin{array}{lll}
t, x \models a & \text{if} & \lambda(x) = a \\
t, x \models \neg\varphi & \text{if} & t, x \not\models \varphi \\
t, x \models \varphi \vee \psi & \text{if} & t, x \models \varphi \text{ or } t, x \models \psi \\
t, x \models \mathsf{EX}\,\varphi & \text{if} & \exists y,\ x \lessdot y \text{ and } t, y \models \varphi \\
t, x \models \varphi\,\mathsf{U}\,\psi & \text{if} & \exists z,\ x \leq z \text{ and } t, z \models \psi \text{ and } t, y \models \varphi, \forall x \leq y < z \\
t, x \models \varphi\,\mathsf{EU}\,\psi & \text{if} & \exists x = y_0 \lessdot \cdots \lessdot y_n,\ \text{with } t, y_n \models \psi \text{ and } t, y_i \models \varphi, \forall 0 \leq i < n
\end{array}
$$

We define $\top = \neg\bot$, hence $\top$ means *true*. We derive some more operators from the above ones. Eventually φ claims the existence of some vertex where φ holds above the current one: $\mathsf{F}\,\varphi = \top\,\mathsf{U}\,\varphi = \top\,\mathsf{EU}\,\varphi$. Its dual operator, always φ, means that φ holds at all positions above the current one: $\mathsf{G}\,\varphi = \neg\,\mathsf{F}\,\neg\varphi$.

The initial formulae $\mathrm{LocTL}_\Sigma(\cdots)$ are defined by the syntax

$$\alpha ::= \bot \mid \mathsf{EM}\,\varphi \mid \neg\alpha \mid \alpha \vee \alpha$$

where $\varphi \in \mathrm{LocTL}^{\mathrm{i}}_\Sigma(\cdots)$. Intuitively, $\mathsf{EM}\,\varphi$ means that φ holds at some minimal vertex. Formally, the semantics is given by

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

The dual $\mathsf{AM}\,\varphi = \neg\,\mathsf{EM}\,\neg\varphi$ means that φ holds for all minimal vertices. One can show that each initial formula is equivalent to a finite disjunction of formulae of the form $\mathsf{AM}\,\varphi \wedge \bigwedge_{a\in A} \mathsf{EM}\,a$ with $\varphi \in \mathrm{LocTL}^{\mathrm{i}}_\Sigma(\cdots)$ and $A \subseteq \Sigma$.

An initial formula $\alpha \in \mathrm{LocTL}_\Sigma(\cdots)$ defines the language $\mathcal{L}(\alpha) = \{t \in \mathbb{R} \mid t \models \alpha\}$ and we say that a trace language $L \subseteq \mathbb{R}$ is expressible in $\mathrm{LocTL}_\Sigma(\cdots)$ if there exists an initial formula $\alpha \in \mathrm{LocTL}_\Sigma(\cdots)$ such that $L = \mathcal{L}(\alpha)$. We denote by $\mathrm{LocTL}_{(\Sigma,D)}(\cdots)$ the set of languages over $\mathbb{R}(\Sigma, D)$ that are expressible by some local temporal formula using the modalities $(\cdots)$.

With local temporal formulae, we can express various alphabetic properties.

$$
\begin{array}{lll}
(a \in \min) & = & \mathsf{EM}\,a \\
(a \in \mathrm{alph}) & = & \mathsf{EM}\,\mathsf{F}\,a \\
(a \in \mathrm{alphinf}) & = & \mathsf{EM}(\mathsf{F}\,a \wedge \mathsf{G}(a \Rightarrow \mathsf{EX}\,\mathsf{F}\,a))
\end{array}
$$

It is clear from the semantics of EX and U that all trace languages expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$ are also expressible in $\mathrm{FO}_\Sigma(<)$, and even in $\mathrm{FO}^3_\Sigma(<)$. This is however not true in general for the existential until EU as shown in the next example.

Note that if two traces t_1 and t_2 have the same minimal letters and for all minimal letters the corresponding upper sets in t_1 and t_2 are the same, then the two traces cannot be distinguished by any formula in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U}, \mathsf{EU})$. The following example for such a situation has been taken from Walukiewicz [21]:

$$t_1 = \left(\begin{tikzcd} a \arrow[r] & b \arrow[r] & c \arrow[r] & b & \cdots \\ d \arrow[r] & c \arrow[u] & & & \end{tikzcd}\right) \qquad t_2 = \left(\begin{tikzcd} a \arrow[r] & b \arrow[d] & & & \\ d \arrow[r] & c \arrow[r] & b \arrow[r] & c & \cdots \end{tikzcd}\right)$$

Since t_1 and t_2 are clearly distinguishable in $\mathrm{FO}_\Sigma(<)$, we deduce that a pure future local temporal logic cannot be expressively complete for $\mathrm{FO}_\Sigma(<)$ as soon as (Σ, D) is not a cograph. Note that it is easy to distinguish t_1 from t_2 if we allow some past tense modalities or if we introduce a dummy minimal vertex. One can also show that the language $ad(bc)^* \subseteq \mathbb{M}$ of finite traces is first order but cannot be expressed in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U}, \mathsf{EU})$.

We are interested in the expressive completeness of pure future local temporal logics. The simple example above shows that we can restrict our study to traces defined by a dependence alphabet that are cographs.

The following lemma shows that we can *localize* initial formulae.

Lemma 1. *Let* $\alpha \in \mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U}, \mathsf{EU})$ *be an initial formula. There exists an internal formula* $\mathrm{loc}(\alpha) \in \mathrm{LocTL}^{\mathrm{i}}_\Sigma(\mathsf{EX}, \mathsf{U})$ *such that for all* $t = t_1 t_2 \in \mathbb{R}$ *and* $x \in \max(t_1)$ *with* $\min(t_2) \subseteq D(\lambda(x))$*, we have* $t_2 \models \alpha$ *if and only if* $t, x \models \mathrm{loc}(\alpha)$*. The same holds for the fragment* $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$.

Proof. Clearly, we have $\mathrm{loc}(\alpha \vee \beta) = \mathrm{loc}(\alpha) \vee \mathrm{loc}(\beta)$ and $\mathrm{loc}(\neg\alpha) = \neg\mathrm{loc}(\alpha)$. The interesting case is $\mathsf{EM}\,\varphi$ where $\varphi \in \mathrm{LocTL}^{\mathrm{i}}$. We have, $\mathrm{loc}(\mathsf{EM}\,\varphi) = \mathsf{EX}(\varphi)$. $\square$

4 Universal until

In this section, we consider the fragment of the local temporal logic using next and the *universal* until, only. We give the following characterization of the expressive completeness of $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$ with respect to $\mathrm{FO}_\Sigma(<)$.

Theorem 2. *Let* (Σ, D) *be a dependence alphabet. Then we have the equality* $\mathrm{LocTL}_{(\Sigma,D)}(\mathsf{EX}, \mathsf{U}) = \mathrm{FO}_{(\Sigma,D)}(<)$ *if and only if* (Σ, D) *is a cograph.*

We have seen in the previous section that $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$ is not expressively complete, if (Σ, D) is not a cograph. Conversely, for all dependence alphabets, we already know that $\mathrm{LocTL}_{(\Sigma,D)}(\mathsf{EX}, \mathsf{U}) \subseteq \mathrm{FO}^3_{(\Sigma,D)}(<)$ and that first order languages coincide with aperiodic languages (Theorem 1). Hence, in order to get the converse inclusion, we will prove that $\mathrm{AP} \subseteq \mathrm{LocTL}_{(\Sigma,D)}(\mathsf{EX}, \mathsf{U})$, if (Σ, D) is a cograph.

Before going into the proof, we derive an immediate corollary. It is well-known that for words, all first order languages can be expressed with only 3 first order variables. This result has been extended to traces by Walukiewicz [22]. In the special case of (Σ, D) being a cograph, we get this result as a trivial consequence of Theorem 2.

Corollary 1. *If (Σ, D) is a cograph, then* $\mathrm{FO}_\Sigma(<) = \mathrm{FO}^3_\Sigma(<)$.

If (Σ, D) is a cograph, then either Σ is a singleton, or Σ is the disjoint union of two non empty sets $\Sigma = A \cup B$ with either $A \times B \subseteq I$ or $A \times B \subseteq D$. We consider these three cases in turn.

Assume that $\Sigma = \{a\}$. Then, traces over Σ coincide with words over Σ and the semantics of EX and U coincide with the semantics of the usual *next* and *until* modalities over words. Therefore, by Kamp's classical theorem [11, 7] we obtain the claim in this case.

Next, we consider the case where the alphabet is the disjoint union of two independent subset. This means $\mathbb{R}(\Sigma, D) = \mathbb{R}_A \times \mathbb{R}_B$ is a direct product.

Proposition 1. *Assume that $\Sigma = A \cup B$ with $A \times B \subseteq I$. If we have both $\mathrm{AP}_A \subseteq \mathrm{LocTL}_A(\mathsf{EX}, \mathsf{U})$ and $\mathrm{AP}_B \subseteq \mathrm{LocTL}_B(\mathsf{EX}, \mathsf{U})$, then $\mathrm{AP}_\Sigma \subseteq \mathrm{LocTL}_{(\Sigma,D)}(\mathsf{EX}, \mathsf{U})$.*

The last case and the most interesting one is when the dependence alphabet is the disjoint union of two fully dependent subsets: $\Sigma = A \cup B$ with $A \cap B = \emptyset$ and $A \times B \subseteq D$. In this case, the trace monoid $\mathbb{M}$ is the free product of the monoids $\mathbb{M}_A$ and $\mathbb{M}_B$.

Proposition 2. *Assume that $\Sigma = A \cup B$ with $A \cap B = \emptyset$ and $A \times B \subseteq D$. If we have both $\mathrm{AP}_A \subseteq \mathrm{LocTL}_A(\mathsf{EX}, \mathsf{U})$ and $\mathrm{AP}_B \subseteq \mathrm{LocTL}_B(\mathsf{EX}, \mathsf{U})$, then $\mathrm{AP}_\Sigma \subseteq \mathrm{LocTL}_{(\Sigma,D)}(\mathsf{EX}, \mathsf{U})$.*

Our proof is inspired by a technique introduced by Wilke [23] in order to show that aperiodic languages over words are expressible in LTL. This method was then extended in order to cope with traces: We have shown in [3] that the most natural global temporal logic over traces is expressively complete with respect to $\mathrm{FO}_\Sigma(<)$.

We use several splittings of languages in products. We shall use the following composition lemmas to get the expressibility of the products from the expressibility of their components. We assume until the end of this section that $\Sigma = A \cup B$ with $A \cap B = \emptyset$ and $A \times B \subseteq D$.

Lemma 2. *Let $L \subseteq \mathbb{R}_A$ be a language expressible in $\mathrm{LocTL}_A(\mathsf{EX}, \mathsf{U})$. Then, the language $(L \cap \mathbb{M}_A^+) \cdot (\min \subseteq B)$ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$.*

Lemma 3. *Let $\alpha \in \mathrm{LocTL}_A(\mathsf{EX}, \mathsf{U})$. The language $(\max \subseteq B) \cdot \mathcal{L}_A(\alpha)$ is definable in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$ by the formula*

$$\Big((\mathrm{alph} \subseteq A) \wedge \alpha\Big) \vee \Big((\mathrm{alphinf} \subseteq A) \wedge \mathsf{EM}\big(\mathsf{F}(B \wedge \neg\, \mathsf{EX}\,\mathsf{F}\, B \wedge \mathrm{loc}(\alpha))\big)\Big).$$

Lemma 4. *Let $L \subseteq \mathbb{R}$ be a language expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$. Then, the language $(L \cap (\max \subseteq B)) \cdot \mathbb{R}_A$ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$.*

We consider the set $\Delta = \mathbb{M} \cup ((\text{alphinf} \not\subseteq A) \cap (\text{alphinf} \not\subseteq B))$. Note that Δ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$ since it is characterized by alphabetic information. Each trace $x \in \Delta$ has a unique (finite or infinite) factorization $x = x_1 x_2 \cdots$ in non-empty finite traces alternating $\mathbb{M}_A^+$ and $\mathbb{M}_B^+$.

Let $h : \mathbb{M} \to S$ be a morphism into some finite aperiodic monoid S. Let $T = h(\mathbb{M}^+)$ and let $e : T^* \to S$ be the evaluation morphism. Using the unique factorization of elements $x \in \Delta$, we can define a mapping $\sigma : \Delta \to T^\infty$ by $\sigma(x) = h(x_1)h(x_2)\cdots$. This mapping σ allows to reduce our problem to words over the alphabet T.

Lemma 5. *Let $L \subseteq \mathbb{R}$ be recognized by h. Then, $L \cap \Delta = \sigma^{-1}(K)$ for some language $K \in T^\infty$ expressible in $\mathrm{LTL}_T(\mathsf{XU})$.*

Proof. Let $K = [\sigma(L \cap \Delta)]_{\approx_e}$. By definition, K is recognized by the evaluation morphism e into the aperiodic monoid S. Hence K is an aperiodic word language over T. Since, by Kamp's Theorem on words, $\mathrm{AP}_T = \mathrm{FO}_T(<) = \mathrm{LTL}_T(\mathsf{XU})$, we deduce that K is expressible in $\mathrm{LTL}_T(\mathsf{XU})$. Therefore, it remains to show that $L \cap \Delta = \sigma^{-1}(K)$. The inclusion $L \cap \Delta \subseteq \sigma^{-1}(K)$ is clear.

For the converse inclusion, let $y \in \sigma^{-1}(K)$. There exists $x \in L \cap \Delta$ such that $\sigma(x) \approx_e \sigma(y)$. Note that x is finite iff y is finite and in this case $h(x) = e(\sigma(x)) = e(\sigma(y)) = h(y)$. Therefore, $y \approx_h x \in L$ and we deduce $y \in L$. Assume now that x and y are both infinite. Using the following claim we also have $y \approx_h x$ and we deduce as above that $y \in L$.

Claim: Let $x = x_1 x_2 \cdots \in \mathbb{R}$ and $y = y_1 y_2 \cdots \in \mathbb{R}$ with $x_i, y_i \in \mathbb{M}^+$. If $h(x_1)h(x_2)\cdots \approx_e h(y_1)h(y_2)\cdots$, then $x \approx_h y$. ☐

In order to make use of the previous lemma, we need to lift through σ^{-1} an LTL formula over T to some local temporal formula. This is the purpose of the next lemma.

Lemma 6. *Let $f \in \mathrm{LTL}_T(\mathsf{XU})$. There exists a formula $\widetilde{f} \in \mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$ such that $\mathcal{L}(\widetilde{f}) = \sigma^{-1}(\mathcal{L}(f))$.*

Proof. The lemma is shown by structural induction on the formula f. First, $\widetilde{\bot} = \bot$, $\widetilde{\varphi \vee \psi} = \widetilde{\varphi} \vee \widetilde{\psi}$, and $\widetilde{\neg\varphi} = \Delta \wedge \neg\widetilde{\varphi}$. Now, it can be shown that $\widetilde{\varphi \,\mathsf{XU}\, \psi}$ equals

$$\Delta \wedge \mathsf{EM}\Big(\big((A \wedge \mathsf{EX}\,A) \vee (B \wedge \mathsf{EX}\,B) \vee \mathrm{loc}(\widetilde{\varphi})\big) \,\mathsf{U}\, \big(((A \wedge \mathsf{EX}\,B) \vee (B \wedge \mathsf{EX}\,A)) \wedge \mathrm{loc}(\widetilde{\psi})\big)\Big)$$

It remains to deal with the case $f = s$ with $s \in T$. We have $\mathcal{L}(s) = sT^\infty$ and

$$\sigma^{-1}(sT^\infty) = \Delta \cap \Big((h^{-1}(s) \cap \mathbb{M}_A^+)(\min \subseteq B) \cup (h^{-1}(s) \cap \mathbb{M}_B^+)(\min \subseteq A)\Big).$$

The language $h^{-1}(s) \cap \mathbb{M}_A^+$ is recognized by $h\!\restriction_{\mathbb{M}_A}$ and is therefore an aperiodic language over $\mathbb{R}_A$. Since we have assumed $\mathrm{AP}_A \subseteq \mathrm{LocTL}_A(\mathsf{EX}, \mathsf{U})$, the language $h^{-1}(s) \cap \mathbb{M}_A^+$ is expressible in $\mathrm{LocTL}_A(\mathsf{EX}, \mathsf{U})$. Using Lemma 2, we deduce that

$(h^{-1}(s) \cap \mathbb{M}_A^+)(\min \subseteq B)$ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$. Similarly, $(h^{-1}(s) \cap \mathbb{M}_B^+)(\min \subseteq A)$ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$ and we deduce that $\sigma^{-1}(sT^\infty)$ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$. $\square$

Now, we have all we need in hand to prove the main result of this section.

Proof of Proposition 2 Let $L \subseteq \mathbb{R}$ be recognized by the morphism h. Since $\mathbb{R} = \Delta \cup (\mathrm{alphinf} \subseteq A) \cup (\mathrm{alphinf} \subseteq B)$, we have

$$L = (L \cap \Delta) \cup (L \cap (\mathrm{alphinf} \subseteq A)) \cup (L \cap (\mathrm{alphinf} \subseteq B)).$$

From Lemmas 5 and 6, we deduce that $L \cap \Delta$ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$.

One can show that the language $L \cap (\mathrm{alphinf} \subseteq A)$ is a finite union of products $(L_1 \cap (\max \subseteq B)) \cdot (L_2 \cap \mathbb{R}_A)$, where L_1 and L_2 are languages recognized by h.

Since $(\max \subseteq B) \subseteq \mathbb{M} \subseteq \Delta$, we have $L_1 \cap (\max \subseteq B) = (L_1 \cap \Delta) \cap (\max \subseteq B)$. Now, L_1 is recognized by h and using Lemmas 5 and 6, we deduce that $L_1 \cap \Delta$ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$. Finally, from Lemma 4 we deduce that $(L_1 \cap (\max \subseteq B)) \cdot \mathbb{R}_A$ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$.

Now, L_2 is recognized by h. Hence, $L_2 \cap \mathbb{R}_A$ is recognized by $h \restriction_{\mathbb{M}_A}$ and is therefore an aperiodic language over $\mathbb{R}_A$. Since we have assumed $\mathrm{AP}_A \subseteq \mathrm{LocTL}_A(\mathsf{EX}, \mathsf{U})$, $L_2 \cap \mathbb{R}_A$ is also expressible in $\mathrm{LocTL}_A(\mathsf{EX}, \mathsf{U})$. Using Lemma 3, we deduce that $(\max \subseteq B) \cdot (L_2 \cap \mathbb{R}_A)$ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$.

Finally, the product $(\max \subseteq B) \cdot \mathbb{R}_A$ is unambiguous, hence we have

$$(L_1 \cap (\max \subseteq B)) \cdot (L_2 \cap \mathbb{R}_A) = (L_1 \cap (\max \subseteq B)) \cdot \mathbb{R}_A \cap (\max \subseteq B) \cdot (L_2 \cap \mathbb{R}_A).$$

We deduce that $(L_1 \cap (\max \subseteq B)) \cdot (L_2 \cap \mathbb{R}_A)$ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$. Therefore, $L \cap (\mathrm{alphinf} \subseteq A)$ is expressible in $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$. $\square$

5 Existential until

In this section, we consider the fragment of the local temporal logic using next and the *existential* until only. We prove the following characterization.

Theorem 3. *Let (Σ, D) be a dependence alphabet. Then we have the equality $\mathrm{LocTL}_{(\Sigma,D)}(\mathsf{EX}, \mathsf{EU}) = \mathrm{FO}_{(\Sigma,D)}(<)$ if and only if (Σ, D) is a cograph.*

We have seen in Section 3 that $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{EU})$ is not expressively complete, if (Σ, D) is not a cograph. Conversely, The proofs of Propositions 1 and 2 can be carried out with slight modifications using the existential until instead of the universal until. Therefore, we get $\mathrm{FO}_{(\Sigma,D)}(<) = \mathrm{AP} \subseteq \mathrm{LocTL}_{(\Sigma,D)}(\mathsf{EX}, \mathsf{EU})$ if (Σ, D) is a cograph. The difficulty with the existential until is that we do not get the converse inclusion for free as with the universal until. Indeed, the semantics of existential until is given by a monadic second order formula since it claims the existence of some path in the trace. This MSO formula can be expressed in first order, if (Σ, D) is a cograph. This is certainly not true for arbitrary dependence alphabet as shown by the following example.

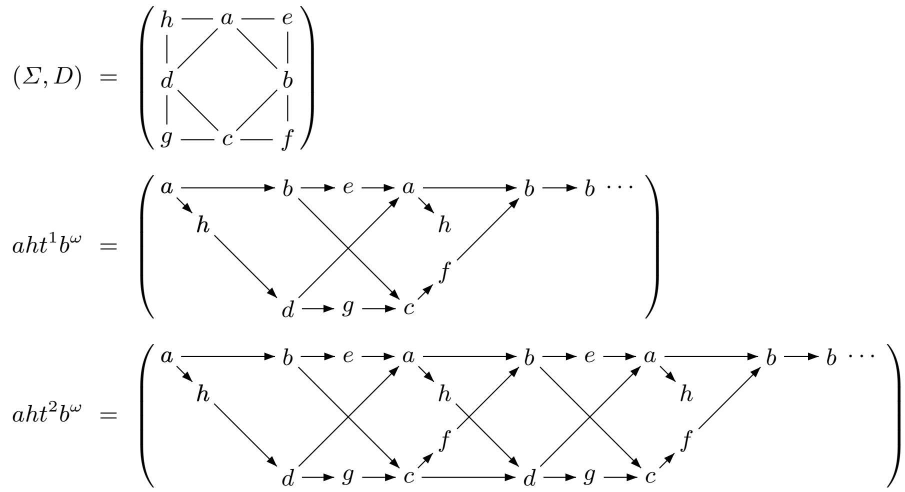


Fig. 1. The dependence alphabet (Σ, D) and some traces.

Consider the dependence alphabet (Σ, D) depicted in Figure 1. Let $t = bdegachf \in \mathbb{M}$ and let $\varphi = (a \vee b \vee c \vee d)\ \mathsf{EU}\ \mathsf{G}\, b$. Then, it is easy to verify that $aht^n b^\omega \models \mathsf{EM}\,\varphi$ if and only if n is even. Now, the trace language t^* is aperiodic. Therefore, the language aht^*b^ω is aperiodic as well and we have $aht^*b^\omega \cap \mathcal{L}(\mathsf{EM}\,\varphi) = ah(t^2)^*b^\omega$ which is not aperiodic. Since aperiodic languages are closed under intersection, we deduce that $\mathcal{L}(\mathsf{EM}\,\varphi)$ is not aperiodic. Therefore, $\mathrm{LocTL}_{(\Sigma,D)}(\mathsf{EU}) \not\subseteq \mathrm{FO}_{(\Sigma,D)}(<)$ for the dependence alphabet of Figure 1.

The key result for showing that $\mathrm{LocTL}_{(\Sigma,D)}(\mathsf{EX}, \mathsf{EU}) \subseteq \mathrm{FO}_{(\Sigma,D)}(<)$, if the dependence alphabet is a cograph, is to express in $\mathrm{FO}_\Sigma(<)$ the existence of a path satisfying some first order formula. Let $\varphi(z)$ be a first order formula with z as only free variable and let $C \subseteq \Sigma$. We define the formula $\mathrm{Path}_C^\varphi(x, y)$ by

$$\begin{aligned}&(\forall x \le z \le y,\ \lambda(z) \in C) \text{ and}\\ &\qquad (\exists x = z_1 \lessdot \cdots \lessdot z_n = y,\ \text{with } \varphi(z_k) \text{ for all } 1 \le k \le n).\end{aligned}$$

Lemma 7. *If (Σ, D) is a cograph, then the formula $\mathrm{Path}_C^\varphi(x, y)$ can be expressed in first order.*

Proof. If (Σ, D) is a cograph, then either Σ is a singleton, or Σ is the disjoint union of two non empty sets $\Sigma = A \cup B$ with either $A \times B \subseteq I$ or $A \times B \subseteq D$. We consider these three cases in turn.

First, assume that $\Sigma = \{a\}$ is a singleton, then

$$\mathrm{Path}_a^\varphi(x, y) = x \le y \wedge \forall x \le z \le y,\ (\lambda(z) = a) \wedge \varphi(z).$$

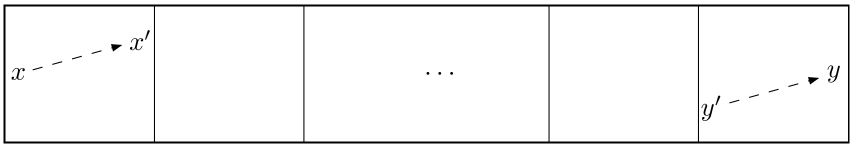
Next, Assume that $\Sigma = A \cup B$ with $A \times B \subseteq I$. Then, we have

$$\mathrm{Path}_{A \cup B}^\varphi(x, y) = \mathrm{Path}_A^\varphi(x, y) \vee \mathrm{Path}_B^\varphi(x, y).$$

Finally, we consider the more interesting case $\Sigma = A \cup B$ with $A \cap B = \emptyset$ and $A \times B \subseteq D$. Then, we have

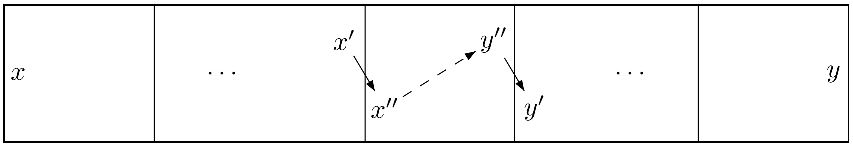
$$\mathrm{Path}^{\varphi}_{A\cup B}(x,y) = \mathrm{Path}^{\varphi}_{A}(x,y) \vee \mathrm{Path}^{\varphi}_{B}(x,y) \vee (\Phi(x,y) \wedge \Psi(x,y))$$

where the formulae $\Phi(x,y)$ and $\Psi(x,y)$ are defined below. We give simultaneously pictures that should help understanding the formulae. The vertical lines in the pictures indicate a separation between factors from $\mathbb{M}_A^+$ and factors from $\mathbb{M}_B^+$.



$$\Phi(x,y) = \exists x \le x' < y' \le y, \begin{pmatrix} \quad \forall x \le u \le x',\ (\lambda(u) \in A \Leftrightarrow \lambda(x) \in A) \\ \wedge\ \forall x' \lessdot u,\ (\lambda(u) \in A \Leftrightarrow \lambda(x) \notin A) \\ \wedge\ (\mathrm{Path}^{\varphi}_{A}(x,x') \vee \mathrm{Path}^{\varphi}_{B}(x,x')) \\ \wedge\ \forall y' \le u \le y,\ (\lambda(u) \in A \Leftrightarrow \lambda(y) \notin A) \\ \wedge\ \forall u \lessdot y',\ (\lambda(u) \in A \Leftrightarrow \lambda(y) \notin A) \\ \wedge\ (\mathrm{Path}^{\varphi}_{A}(y',y) \vee \mathrm{Path}^{\varphi}_{B}(y',y)) \end{pmatrix}$$

and



$$\Psi(x,y) = \forall x \le x' < y' \le y, \begin{pmatrix} \quad (\lambda(x') \in A \Leftrightarrow \lambda(y') \in A) \\ \wedge\ \exists x' < u < y',\ (\lambda(u) \in A \Leftrightarrow \lambda(x') \notin A) \\ \wedge\ \forall x' < u < y',\ (\lambda(u) \in A \Leftrightarrow \lambda(x') \notin A) \end{pmatrix}$$
$$\Rightarrow \left(\exists x' \lessdot x'' \le y'' \lessdot y',\ \ \mathrm{Path}^{\varphi}_{A}(x'',y'') \vee \mathrm{Path}^{\varphi}_{B}(x'',y'')\right)$$

□

From this lemma, we immediately deduce the desired result.

Proposition 3. *If the dependence alphabet is a cograph, then the existential modality* EU *can be expressed in* $\mathrm{FO}_{\Sigma}(<)$.

Proof. Follows directly from the Lemma 7 since we have

$$t, x \models \varphi \,\mathsf{EU}\, \psi \ \text{ iff } \ t, x \models \psi \text{ or } \exists y, \mathrm{Path}^{\varphi}_{\Sigma}(x,y) \text{ and } \exists z,\ y \lessdot z \text{ and } t, z \models \psi$$

The logic TLC introduced in [1] uses EX, EU, the dual past modalities EY and ES and two additional modalities $\mathsf{Eco}\,\varphi$ claiming that φ holds for some vertex that is concurrent with the current one; and $\mathsf{EG}\,\varphi$ claiming the existence of some maximal path starting from the current vertex such that φ holds everywhere along this path. EY and Eco are clearly first order modalities, while ES and EG are in general only (monadic) second order. But, using the technique of Lemma 7 and Proposition 3 one can show that ES and EG are expressible in first order, if the dependence alphabet is a cograph.

Proposition 4. *Let (Σ, D) be a dependence alphabet which is a cograph. Then,* $\mathrm{TLC}(\Sigma, D) \subseteq \mathrm{FO}_{(\Sigma,D)}(<)$.

6 Complexity

In this section, we show that the satisfiability problem for local temporal logics is PSPACE-complete. The PSPACE-hardness is a consequence of the PSPACE-hardness for words. For TLC the inclusion in PSPACE has been shown in [1]. In order to prove that our problem is still in PSPACE, we have to deal with the universal until–operator which we have introduced here. For this, we associate with each initial formula α an alternating automaton that accepts all the linearizations of the traces that model α. Here, we describe the construction for the pure future local temporal logic $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$. We assume the reader is familiar with alternating automata and the usual translation from LTL formulae to alternating automata [19].

Let $\alpha \in \mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$. Without loss of generality, we assume that the negations in α are only over formulae of the form $\mathsf{EM}\,\varphi$ or b or $\mathsf{EX}\,\varphi$ or $\varphi \,\mathsf{U}\, \psi$. We construct an alternating automaton $\mathcal{A}_\alpha$ as follows. The set of states is $Q \cup \overline{Q}$ where

$$\begin{aligned}
Q = \{\xi \mid \xi & \text{ is a subformula of } \alpha \text{ of the form } \mathsf{EM}\,\varphi \text{ or } b \text{ or } \mathsf{EX}\,\varphi \text{ or } \varphi \,\mathsf{U}\, \psi\} \\
\cup \{(D(B), \mathsf{EM}\,\varphi) & \mid \mathsf{EM}\,\varphi \text{ is a subformula of } \alpha,\ B \subseteq \Sigma \\
& \quad \text{and } \Sigma \setminus D(B) \neq \emptyset\} \\
\cup \{(D(a), D(B), \mathsf{EX}\,\varphi) & \mid \mathsf{EX}\,\varphi \text{ is a subformula of } \alpha,\ a \in \Sigma,\ B \subseteq \Sigma, \\
& \quad \{a\} \cup B \text{ connected and } D(a) \setminus D(B) \neq \emptyset\} \\
\cup \{(D(A), D(B), \varphi \,\mathsf{U}\, \psi) & \mid \varphi \,\mathsf{U}\, \psi \text{ is a subformula of } \alpha,\ A, B \subseteq \Sigma, \\
& \quad A \cup B \text{ connected and } D(A) \setminus D(B) \neq \emptyset\}
\end{aligned}$$

and $\overline{Q} = \{\neg p \mid p \in Q\}$. The set of positive boolean combinations of states is denoted $\mathbb{B}^+(Q \cup \overline{Q})$. The initial state is $\alpha \in \mathbb{B}^+(Q \cup \overline{Q})$. We define the extended transition function $\delta : \mathbb{B}^+(Q \cup \overline{Q}) \times \Sigma \to \mathbb{B}^+(Q \cup \overline{Q})$ as follows.

$$\delta(f \vee g, a) = \delta(f, a) \vee \delta(g, a), \text{ for } f, g \in \mathbb{B}^+(Q).$$

$$\delta(f \wedge g, a) = \delta(f, a) \wedge \delta(g, a), \text{ for } f, g \in \mathbb{B}^+(Q).$$

$$\delta(\mathsf{EM}\,\varphi, a) = \begin{cases} \quad \delta(\varphi, a) & \\ \vee\ (D(a), \mathsf{EM}\,\varphi) & \text{if } D(a) \neq \Sigma. \end{cases}$$

$$\delta((D(B), \mathsf{EM}\,\varphi), a) = \begin{cases} \quad (D(B) \cup D(a), \mathsf{EM}\,\varphi) & \text{if } D(B) \cup D(a) \neq \Sigma \\ \vee\ \delta(\varphi, a) & \text{if } a \notin D(B). \end{cases}$$

$$\delta(b, a) = \begin{cases} \top & \text{if } a = b, \\ \bot & \text{if } a \neq b. \end{cases}$$

$$\delta(\mathsf{EX}\,\varphi, a) = (D(a), \emptyset, \mathsf{EX}\,\varphi).$$

$$\delta((D(A), D(B), \mathsf{EX}\,\varphi), a) = \begin{cases} \quad (D(A), D(B) \cup D(a), \mathsf{EX}\,\varphi) & \text{if } a \in D(A) \cup D(B) \text{ and} \\ & \quad D(A) \setminus (D(B) \cup D(a)) \neq \emptyset \\ \vee\ (D(A), D(B), \mathsf{EX}\,\varphi) & \text{if } a \notin D(A) \cup D(B), \\ \vee\ \delta(\varphi, a) & \text{if } a \in D(A) \setminus D(B). \end{cases}$$

$$\delta(\varphi \,\mathsf{U}\, \psi, a) = \delta(\psi, a) \vee (\delta(\varphi, a) \wedge (D(a), \emptyset, \varphi \,\mathsf{U}\, \psi)).$$

$$\delta((D(A), D(B), \varphi \,\mathsf{U}\, \psi), a) = \begin{cases} \quad (D(A), D(B) \cup D(a), \varphi \,\mathsf{U}\, \psi) & \text{if } a \in D(A) \cup D(B) \text{ and} \\ & \quad D(A) \setminus (D(B) \cup D(a)) \neq \emptyset \\ \vee\ (D(A), D(B), \varphi \,\mathsf{U}\, \psi) & \text{if } a \notin D(A) \cup D(B), \\ \vee\ \delta(\varphi, a) \wedge (D(A) \cup D(a), D(B), \varphi \,\mathsf{U}\, \psi) & \text{if } a \in D(A) \setminus D(B), \\ \vee\ \delta(\psi, a) & \text{if } a \in D(A) \setminus D(B). \end{cases}$$

$$\delta(\neg p, a) = \widetilde{\delta(p, a)}, \text{ for } p \in Q;$$

where $\widetilde{f \vee g} = \widetilde{f} \wedge \widetilde{g}$ and $\widetilde{f \wedge g} = \widetilde{f} \vee \widetilde{g}$, for $f, g \in \mathbb{B}^+(Q)$; and $\widetilde{p} = \neg p$ and $\widetilde{\neg p} = p$ for $p \in Q$. The actual transition function of $\mathcal{A}_\alpha$ is the restriction of δ to $(Q \cup \overline{Q}) \times \Sigma$.

The states in Q represent obligations that the word has to fulfill in order to be accepted, hence an infinite branch looping on such a state should not be accepted. Therefore, the co-Büchi acceptance condition is given by the set Q. Note that the alternating automaton defined above is *very weak*. Hence, each infinite branch in the run of the alternating automaton is ultimately constant and the co-Büchi acceptance condition Q is equivalent with the Büchi acceptance condition $\overline{Q}$.

Proposition 5. *The automaton $\mathcal{A}_\alpha$ accepts the word language*

$$\mathcal{L}(\mathcal{A}_\alpha) = \{w \in \Sigma^\infty \mid [w] \models \alpha\},$$

where $[w]$ denotes the trace associated with the word $w \in \Sigma^\infty$.

Let $N(\Sigma, D) = 1 + |\{(D(A), D(B)) \mid A, B \subseteq \Sigma, A \cup B$ connected and $D(A) \setminus D(B) \neq \emptyset\}|$. The number of states of the automaton $\mathcal{A}_\alpha$ is $|Q \cup \overline{Q}| \leq 2N(\Sigma, D)|\alpha|$. Note that when the dependence relation is full, that is, when traces are actually words, then $N(\Sigma, D) = 2$ and the size of our automaton does not depend on the size of the alphabet. In this case, we essentially get the usual construction for LTL formulae over words.

In fact, we shall use a slightly better parameter. Decompose (Σ, D) into its connected components such that (Σ, D) is a disjoint union of connected graphs (Σ_i, D_i) for $1 \leq i \leq k$. Let $M = \max_{1 \leq i \leq k} N(\Sigma_i, D_i)$. Using this parameter M we can state.

Theorem 4. *Let M be defined as above and let M be bounded by some constant which is not part of the input. Then the satisfiability problem for the local logic* $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$ *is PSPACE-complete.*

Proof. The PSPACE-hardness follows from the word case. The PSPACE algorithm reduces the satisfiability problem in a first phase to a conjunction of satisfiability problems, one for each connected component of (Σ, D). Then we can check emptiness for the alternating automata according to the construction in Proposition 5. Checking emptiness for an alternating automaton can be done in PSPACE with respect to the size the automaton. $\square$

The construction of an alternating automaton associated with a local temporal logic formula can be carried out for the *existential* until EU and the operator EG from TLC. Also, we can extend the construction for the past modalities EY, S and ES and for the operator Eco from TLC is we use two-ways alternating automata. Since the emptiness problem for two-ways alternating automata is also PSPACE-complete [10], we get a similar result for the local temporal logic using all operators. This extends the result in [1] concerning TLC.

7 Conclusion

We have defined a basic and natural local logic for Mazurkiewicz traces which is expressively complete with respect to first order if and only if the dependence alphabet is a cograph, i.e., all traces are series parallel graphs. The main open problem remains to define a (natural) local logic which yields expressive completeness for more general (best for all) dependence alphabets, and such that the satisfiability problem is in PSPACE or at least elementary.

There is a proposal by Walukiewicz [21] for a local logic for traces, but his focus is on monadic second order logic and based on a μ–calculus, so it is of quite different spirit.

References

1. R Alur, D. Peled, and W. Penczek. Model-checking of causality properties. In *Proceedingss of the Tenth IEEE Symposium on Logic in Computer Science (LICS'95)*, pages 90–100, 1995.
2. D.G. Corneil, H. Lerchs, and L.S. Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3:163–174, 1981.
3. V. Diekert and P. Gastin. LTL is expressively complete for Mazurkiewicz traces. In *Proceedings of ICALP 2000*, number 1853 in Lecture Notes in Computer Science, pages 211–222. Springer Verlag, 2000.
4. V. Diekert and G. Rozenberg, editors. *The Book of Traces.* World Scientific, Singapore, 1995.
5. W. Ebinger and A. Muscholl. Logical definability on infinite traces. *Theoretical Computer Science*, 154:67–84, 1996.
6. W. Ebinger. *Charakterisierung von Sprachklassen unendlicher Spuren durch Logiken.* Dissertation, Institut für Informatik, Universität Stuttgart, 1994.
7. D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *Conference Record of the 12th ACM Symposium on Principles of Programming Languages*, pages 163–173, Las Vegas, Nev., 1980.

8. J. G. Henriksen. An expressive extension of TLC. In *Proceedings of ASIAN'99*, number 1742 in Lecture Notes in Computer Science, pages 126–138. Springer Verlag, 1999.
9. J. G. Henriksen. *Logic and Automata for Verification: Expressiveness and Decidability Issues*. PhD thesis, Dept. of Comp. Sci., University of Aarhus, 2000.
10. T. Jiang and B. Ravikumar. A note on the space complexity of some decision problems fir finite automata. *Information Processing Letters*, 40:25–31, 1991.
11. J. A. W. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, Calif., 1968.
12. A. Mazurkiewicz. Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University, Aarhus, 1977.
13. M. Mukund and P. S. Thiagarajan. Linear time temporal logics over Mazurkiewicz traces. In *Proceedings of the 21th MFCS, 1996*, number 1113 in Lecture Notes in Computer Science, pages 62–92. Springer Verlag, 1996.
14. P. Niebert. A ν-calculus with local views for sequential agents. In *Proceedings of the 20th MFCS, 1995*, number 969 in Lecture Notes in Computer Science, pages 563–573. Springer Verlag, 1995.
15. W. Penczek. Temporal logics for trace systems: On automated verification. *International Journal of Foundations of Computer Science*, 4:31–67, 1993.
16. R. Ramanujam. Locally linear time temporal logic. In *Proceedings of LICS'96*, Lecture Notes in Computer Science, pages 118–128, 1996.
17. P.S. Thiagarajan and I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. In *Proceedings of LICS'97*, 1997.
18. P. S. Thiagarajan. A trace based extension of linear time temporal logic. In *Proceedings of LICS'94*, pages 438–447, 1994.
19. M. Vardi. An automata-theoretic approach to linear temporal logic. In *Logics for Concurrency: Structure versus Automata*, number 1043 in Lecture Notes in Computer Science, pages 238–266. Springer Verlag, 1996.
20. I. Walukiewicz. Difficult configurations – on the complexity of LTrL. In *Proceedings of ICALP'98*, number 1443 in Lecture Notes in Computer Science, pages 140–151. Springer Verlag, 1998.
21. I. Walukiewicz. Local logics for traces. *Journal of Automata, Languages and Combinatorics*, 2001. To appear.
22. I. Walukiewicz. Private communication, 2001.
23. Th. Wilke. Classifying discrete temporal properties. In *Proceedings of STACS'99*, number 1563 in Lecture Notes in Computer Science, pages 32–46. Springer Verlag, 1999.