Satisfiability and model checking for MSO-definable temporal logics are in PSPACE

Paul Gastin[1] and Dietrich Kuske[2]

[1] LIAFA, Université Paris 7, 2, place Jussieu, F-75251 Paris Cedex 05, Paul.Gastin@liafa.jussieu.fr
[2] Institut für Algebra, TU Dresden, D-01062 Dresden, kuske@math.tu-dresden.de

Abstract. Temporal logics over Mazurkiewicz traces have been extensively studied over the past fifteen years. In order to be usable for the verification of concurrent systems they need to have reasonable complexity for the satisfiability and the model checking problems. Whenever a new temporal logic was introduced, a new proof (usually non trivial) was needed to establish the complexity of these problems. In this paper, we introduce a unified framework to define local temporal logics over traces. We prove that the satisfiability problem and the model checking problem for asynchronous Kripke structures for local temporal logics over traces are decidable in PSPACE. This subsumes and sometimes improves all complexity results previously obtained on local temporal logics for traces.

1 Introduction

Over the past fifteen years, a lot of papers have been devoted to the study of temporal logics over partial orders and in particular over Mazurkiewicz traces. This is motivated by the need for specification languages that are suited for concurrent systems where a property should not depend on the ordering between independent events. Hence logics over linearizations of behaviors are not adequate and logics over partial orders were developed. In order to be useful for the verification of concurrent systems, these specification languages should enjoy reasonable complexity for the satisfiability and the model checking problems.

Temporal logics over traces can be classified in *global* ones and *local* ones. Here we are interested in the latter. They are evaluated at single events corresponding to local views of processes. Process based logics [13, 14, 11] were introduced by Thiagarajan and shown to be decidable in EXPTIME using difficult results on *gossip* automata. A specific feature of process based logics is the until modality that can only walk along a single process. Another approach was taken in [1] were the until is existential and walks along some path in the Hasse diagram of the partial order. The decidability in PSPACE of this logic was shown using a tableau construction. Due to this existential until, this logic is not contained in first order logic of traces [4]. In the quest for an expressively complete local temporal logic over traces, a universal until was introduced in [4] and filtered variants together with past modalities were needed in [7]. Again these logics were proved to be decidable in PSPACE using alternating automata. For each local logic, a specific proof has to be developed for the complexity of the satisfiability or the model checking problem. Such proofs are usually difficult and span over several pages.

In this paper, we introduce a unified framework to define local temporal logics over traces (Section 5). This approach is inspired from [12]. Basically, a local temporal logic is given by a finite set of modalities whose semantics is given by a monadic second order (MSO) formula having a single individual free variable. We call these logics MSO-definable. We show that all local temporal logics considered so far (and much more) are MSO-definable. Then we show that the satisfiability problem and the model checking problem for asynchronous Kripke structures for MSO-definable temporal logics over traces are decidable in PSPACE (Section 6). This subsumes and sometimes improves all the complexity results over local logics discussed above. We would like to stress that the proofs for our main results are actually simpler than some proofs specific to some local logics and even from a practical point of view, our decision procedures are as efficient as specific ones could be. Also, our results may be surprising at first since the satisfiability problem for MSO is non elementary, but because we use a finite set of MSO-definable modalities our decision problems stay in PSPACE.

Actually, we start by introducing our MSO-definable temporal logics for words (Section 3) and we prove that the satisfiability and the model checking problems are decidable in PSPACE (Section 4). Though words are special cases of traces, we believe that the paper is easier to follow in this way and that results for words are interesting by themselves. A reader that is not familiar with traces can easily understand the results for words. Other general frameworks for temporal logics over words have been studied [17, 16, 9]. In [17] the modalities are defined by right linear grammars extended to infinite words while in [16, 9] the modalities are defined by various kinds of automata (either non-deterministic Büchi, or alternating or two-way alternating). Note that in these approaches, the automata that define the modalities are part of the formulas. In all cases, the satisfiability problem is proved to be decidable in PSPACE. Our approach is indeed similar but differs by the way modalities are defined. We have chosen MSO modalities because this is how the semantics of local temporal logics over traces is usually defined. In this way, we trivially obtain as corollaries of our main theorems the complexity results for local temporal logics over traces. It is also possible to give automata for the local modalities over traces and apply the results of [16, 9]. This is basically what is done in [5] but such a reduction is difficult and long.

2 Monadic second order logic

Let Σ be an alphabet. *Monadic second order logic* (MSO) is a formalism to speak about the properties of words over Σ. It is based on individual variables $x, y, z, \ldots$ that range over positions in the word (*i.e.*, over elements of $\mathbb{N}$) and on set variables $X, Y, Z, \ldots$ that range over sets of positions (*i.e.*, over subsets of $\mathbb{N}$). Its atomic formulas are $x \leq y$, $P_a(x)$ for $a \in \Sigma$ and $X(x)$ where x, y are individual variables and X is a set variable. The use of Boolean connectives $\wedge, \vee, \neg, \rightarrow$ *etc* and quantification $\exists x$ and $\exists X$ over individual and set variables allows to build more complex formulas. We denote by $\mathrm{MSO}_\Sigma(<)$ the set of MSO formulas over the alphabet Σ.

To define the semantics of a formula, let $w = w_0 w_1 \cdots \in \Sigma^\infty = \Sigma^+ \cup \Sigma^\omega$. We denote by $|w|$ the length of w which may be finite or infinite. A *position* in w is an integer p with $0 \leq p < |w|$. A *valuation* in w for the formula φ is a mapping ν that assigns positions in w to the free individual variables of φ and sets of positions in w to the free set variables of φ.

$$\begin{array}{ll}
w, \nu \models_{\mathrm{MSO}} x \leq y & \text{if } \nu(x) \leq \nu(y) \\
w, \nu \models_{\mathrm{MSO}} P_a(x) & \text{if } w_{\nu(x)} = a \\
w, \nu \models_{\mathrm{MSO}} X(x) & \text{if } \nu(x) \in \nu(X) \\
w, \nu \models_{\mathrm{MSO}} \exists x \varphi & \text{if } w, \nu[x \mapsto p] \models_{\mathrm{MSO}} \varphi \text{ for some position } p \text{ in } w \\
w, \nu \models_{\mathrm{MSO}} \exists X \varphi & \text{if } w, \nu[X \mapsto P] \models_{\mathrm{MSO}} \varphi \text{ for some set } P \text{ of positions in } w
\end{array}$$

Here, $\nu[x \mapsto p]$ is the mapping that coincides with ν except for the value of x which is p; $\nu[X \mapsto P]$ is defined similarly. If φ is an MSO formula with free variables $X_1, \ldots, X_\ell, x_1, \ldots, x_k$ and ν is a valuation in a word w then we also write $w \models_{\mathrm{MSO}} \varphi(\nu(X_1), \ldots, \nu(X_\ell), \nu(x_1), \ldots, \nu(x_k))$ for $w, \nu \models_{\mathrm{MSO}} \varphi$.

3 A uniform framework for temporal logics over words

We introduce our approach on an example. We use PLTL (linear temporal logic with past) because it is well-known and allows us to introduce easily the main definitions. We start with a finite alphabet Σ and recall that the syntax of PLTL is given by

$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \mathsf{Y}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \varphi\,\mathsf{S}\,\varphi$$

where a ranges over Σ. We assume the reader is familiar with the semantics of PLTL over words: $w, p \models_{\mathrm{PLTL}} \varphi$ means that the formula φ holds in the word w at position p. Here $w = w_0 w_1 \cdots \in \Sigma^\infty$ and $0 \leq p < |w|$. For instance,

$$\begin{array}{lll}
w, p \models_{\mathrm{PLTL}} a & \text{if} & w_p = a \\
w, p \models_{\mathrm{PLTL}} \mathsf{Y}\,\varphi & \text{if} & p > 0 \text{ and } w, p-1 \models_{\mathrm{PLTL}} \varphi \\
w, p \models_{\mathrm{PLTL}} \varphi\,\mathsf{U}\,\psi & \text{if} & \exists k (p \leq k \text{ and } w, k \models_{\mathrm{PLTL}} \psi \text{ and} \\
& & \qquad w, j \models_{\mathrm{PLTL}} \varphi \text{ for all } p \leq j < k
\end{array}$$

In order to define PLTL in our framework, we start with a vocabulary B of modality names and a mapping arity : $B \to \mathbb{N}$ giving the arity of each modality. The modality names of arity 0 are the atomic formulas of $\mathrm{TL}(B)$. Other formulas are obtained from atomic formulas by the application of modalities. The syntax of the temporal logic $\mathrm{TL}(B)$ based on the vocabulary B is then

$$\varphi ::= \sum_{M \in B} M(\underbrace{\varphi, \ldots, \varphi}_{\mathrm{arity}(M)}).$$

For PLTL we consider $B_{PLTL} = \Sigma \cup \{\neg, \mathsf{X}, \mathsf{Y}, \vee, \mathsf{U}, \mathsf{S}\}$ and the arity is 0 for elements in Σ, 1 for $\neg, \mathsf{X}, \mathsf{Y}$ and 2 for $\vee, \mathsf{U}, \mathsf{S}$. The syntax of $\mathrm{TL}(B_{PLTL})$ is then precisely that of $PLTL$.

In order to define the semantics of $\mathrm{TL}(B)$ we consider a mapping $[\![-]\!] : B \to \mathrm{MSO}_\Sigma(<)$ in such a way that if $M \in B$ is of arity ℓ then $[\![M]\!]$ is an ℓ-ary MSO modality, that is, an MSO formula with ℓ free set variables $X_1, \dots, X_\ell$ and one free individual variable x. The intuition is that a word w at position p satisfies $M(\varphi_1, \dots, \varphi_\ell)$ if $w, \nu \models_{\mathrm{MSO}} [\![M]\!](X_1, \dots, X_\ell, x)$ when $\nu(x) = p$ and for each i, $\nu(X_i)$ is the set of positions in w where φ_i holds. For PLTL, the mapping $[\![-]\!]$ is given by

$$\begin{array}{lll}
[\![a]\!](x) & = & P_a(x) \quad \text{for } a \in \Sigma \\
[\![\neg]\!](X_1, x) & = & \neg X_1(x) \\
[\![\mathsf{X}]\!](X_1, x) & = & X_1(x+1) \ = \ \exists z(x < z \wedge X_1(z) \wedge \forall y(x < y \to z \leq y)) \\
[\![\mathsf{Y}]\!](X_1, x) & = & X_1(x-1) \ = \ \exists z(z < x \wedge X_1(z) \wedge \forall y(y < x \to y \leq z)) \\
[\![\vee]\!](X_1, X_2, x) & = & X_1(x) \vee X_2(x) \\
[\![\mathsf{U}]\!](X_1, X_2, x) & = & \exists z(x \leq z \wedge X_2(z) \wedge \forall y(x \leq y < z \to X_1(y))) \\
[\![\mathsf{S}]\!](X_1, X_2, x) & = & \exists z(z \leq x \wedge X_2(z) \wedge \forall y(z < y \leq x \to X_1(y)))
\end{array}$$

Finally, given a word $w \in \Sigma^\infty$ and a formula $\varphi \in \mathrm{TL}(B)$, we define inductively the set φ^w of position in w where φ holds. If $\varphi = M(\varphi_1, \dots, \varphi_\ell)$ where $M \in B$ is of arity $\ell \geq 0$, then

$$\varphi^w = \{p < |w| \mid w \models_{\mathrm{MSO}} [\![M]\!](\varphi_1^w, \dots, \varphi_\ell^w, p)\}.$$

Proposition 1. *Let $\varphi \in \mathrm{TL}(B_{PLTL}) = PLTL$ and $w \in \Sigma^\infty$. Then,*

$$\varphi^w = \{p < |w| \mid w, p \models_{\mathrm{PLTL}} \varphi\}.$$

The proof of this proposition is easy and omitted. What is interesting is that it exhibits an alternative definition of PLTL using a vocabulary B (with arity) and a semantic map $[\![-]\!]$. By varying the vocabulary and the semantic map we have a very general way to define temporal logics for words and therefore a formal framework to state complexity results for a large class of temporal logics. This is exactly what we were looking for.

For convenience, we summarize below the definition of an MSO temporal logics over words.

Definition 2. *We start with a set B consisting of modality names together with a mapping* arity $: B \to \mathbb{N}$ *giving the arity of each modality. Then the syntax of the temporal logic* $\mathrm{TL}(B)$ *is defined by the grammar*

$$\varphi ::= \sum_{M \in B} M(\underbrace{\varphi, \dots, \varphi}_{\mathrm{arity}(M)}).$$

Consider a mapping $[\![-]\!] : B \to \mathrm{MSO}_\Sigma(<)$ such that $[\![M]\!]$ is an ℓ-ary MSO modality, that is, an MSO formula with ℓ free set variables $X_1, \dots, X_\ell$ and one *free individual variable x. Given a word $w \in \Sigma^\infty$ and a formula $\varphi \in \mathrm{TL}(B)$, the semantics is given by the set φ^w of position in w where φ holds. The inductive definition is as follows. If $\varphi = M(\varphi_1, \dots, \varphi_\ell)$ where $M \in B$ is of arity $\ell \geq 0$, then*

$$\varphi^w = \{p < |w| \mid w \models_{\mathrm{MSO}} [\![M]\!](\varphi_1^w, \dots, \varphi_\ell^w, p)\}.$$

We also write $w, p \models \varphi$ for $p \in \varphi^w$.

If we fix the triple (B,arity, $[\![-]\!]$) once for ever, the expressive power of $\mathrm{TL}(B)$ is limited. For instance, the expressive power of PLTL is known to be strictly weaker than that of monadic second order logic [8]. We can extend its expressive power introducing a new modality name **even** of arity 1 with associated MSO-modality

$$[\![\mathbf{even}]\!] = (\exists Y (|Y| \text{ is even } \wedge \forall y (Y(y) \leftrightarrow (X_1(y) \wedge y \geq x)))).$$

The formula $\mathbf{even}(a) \in \mathrm{TL}(\{\mathbf{even}, a\})$ is satisfied by a word w in position p if and only if the word w contains an even number of occurrences of the letter a to the right of p. Recall that this property is not expressible in PLTL [8].

4 Complexity of temporal logics for words

In this section, we show that, whatever the finite set B of modality names and associated MSO-modalities is, the satisfiability and the model checking problems for $\mathrm{TL}(B)$ are decidable in PSPACE.

Satisfiability problem for $\mathrm{TL}(B)$ *over words:* Given a formula $\xi \in \mathrm{TL}(B)$, does there exist a word $w \in \Sigma^\infty$ and a position p in w such that $w, p \models \xi$?

Remark 3. One may also consider *initial* satisfiability of a given formula $\xi \in \mathrm{TL}(B)$, i.e., does there exists a word $w \in \Sigma^\infty$ such that $w, 0 \models \xi$. This problem can be easily reduced to the general satisfiability. Add a modality name init of arity 1 to B with associated MSO-modality $[\![\mathrm{init}]\!](X_1, x) = \exists y (y \leq x \wedge X_1(y) \wedge y \text{ minimal})$. Now, a formula $\xi \in \mathrm{TL}(B)$ is *initially* satisfiable if and only if the formula $\mathrm{init}(\xi)$ is satisfiable.

For a word $w = a_0 a_1 \cdots \in \{0,1\}^\infty$, let $\mathrm{supp}(w) = \{p < |w| \mid a_p = 1\}$ denote the *support* of w. For $\ell \in \mathbb{N}$, we consider the alphabet $\Sigma_\ell = \Sigma \times \{0,1\}^\ell$. A letter $a \in \Sigma_\ell$ will be written $a = (a_0, a_1, \dots, a_\ell)$ and a word $w \in \Sigma_\ell^\infty$ will be identified with a tuple of words of same length in the obvious way: $w = (w_0, w_1, \dots, w_\ell) \in \Sigma^\infty \times (\{0,1\}^\infty)^\ell$ with $|w| = |w_i|$ for $0 \leq i \leq \ell$.

Recall the following result that can easily be extracted from the proof of Büchi's theorem.

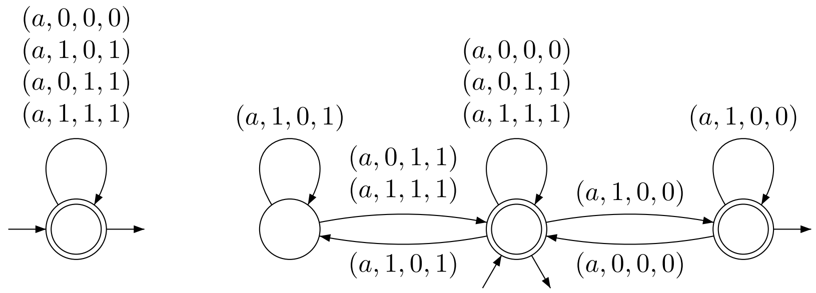
Theorem 4 ([2]). *Let M be an ℓ-ary modality name and $[\![M]\!]$ its associated MSO-modality. Then there exists a Büchi-automaton $\mathcal{B}_M$ over the alphabet $\Sigma_{\ell+1}$ such that $w = (w_0, w_1, \dots, w_{\ell+1}) \in \mathcal{L}(\mathcal{B}_M)$ if and only if* $\mathrm{supp}(w_{\ell+1}) = \{p < |w| \mid w_0 \models_{\mathrm{MSO}} [\![M]\!](\mathrm{supp}(w_1), \dots, \mathrm{supp}(w_\ell), p)\}$.

Proof. Consider the MSO formula

$$\overline{[\![M]\!]}(X_1,\ldots,X_{\ell+1}) = \forall x(X_{\ell+1}(x) \leftrightarrow [\![M]\!](X_1,\ldots,X_\ell,x)).$$

From the proof of Büchi's theorem (see e.g. [15]), we find an automaton $\mathcal{B}_M$ over $\Sigma_{\ell+1}$ such that a word $w = (w_0, w_1, \ldots, w_{\ell+1}) \in \mathcal{L}(\mathcal{B}_M)$ if and only if $w_0 \models_{\mathrm{MSO}} \overline{[\![M]\!]}(\mathrm{supp}(w_1),\ldots,\mathrm{supp}(w_{\ell+1}))$. This is equivalent with $\mathrm{supp}(w_{\ell+1}) = \{p < |w| \mid w_0 \models_{\mathrm{MSO}} [\![M]\!](\mathrm{supp}(w_1),\ldots,\mathrm{supp}(w_\ell),p)\}$ by definition of $\overline{[\![M]\!]}$. $\square$

As examples, we give the automata $\mathcal{B}_\vee$ and $\mathcal{B}_{\mathsf{U}}$:



For formulas φ and ψ, we write $\varphi \le \psi$ if φ is a subformula of ψ (this includes the case $\varphi = \psi$). Let ξ be a formula from $\mathrm{TL}(B)$ and let $\mathrm{Sub}(\xi) = \{\varphi \in \mathrm{TL}(B) \mid \varphi \le \xi\}$. In the sequel, we will consider words over the alphabet $\overline{\Sigma} = \Sigma \times \{0,1\}^{\mathrm{Sub}(\xi)}$. Typically, the elements of $\overline{\Sigma}$ are of the form $\overline{a} = (a, (a_\varphi)_{\varphi\le\xi})$ with $a \in \Sigma$ and $a_\varphi \in \{0,1\}$ for $\varphi \le \xi$. As above, we identify a word $\overline{w} \in \overline{\Sigma}^\infty$ with a tuple of words of same length: $\overline{w} = (w, (w_\varphi)_{\varphi\le\xi})$ with $w \in \Sigma^\infty$, $w_\varphi \in \{0,1\}^\infty$ for $\varphi \le \xi$ and $|\overline{w}| = |w| = |w_\varphi|$.

Now let $\psi = M(\varphi_1,\ldots,\varphi_\ell) \le \xi$. Then $\overline{a}{\restriction}\psi := (a, a_{\varphi_1},\ldots,a_{\varphi_\ell},a_\psi) \in \Sigma_{\ell+1}$. Accordingly, for $\overline{w} \in \overline{\Sigma}^\infty$ we let $\overline{w}{\restriction}\psi = (w, w_{\varphi_1},\ldots,w_{\varphi_\ell},w_\psi) \in \Sigma_{\ell+1}^\infty$.

The construction. For a formula $\varphi \in \mathrm{TL}(B)$, let $\mathrm{top}(\varphi)$ be the outermost modality name of φ. Let $Q = \prod_{\varphi\le\xi} Q_{\mathrm{top}(\varphi)}$ be the set of states of the automaton $\mathcal{A}_\xi$ where $Q_{\mathrm{top}(\varphi)}$ is the set of states of the Büchi-automaton $\mathcal{B}_{\mathrm{top}(\varphi)}$. The alphabet of $\mathcal{A}_\xi$ is $\overline{\Sigma}$. For a letter $\overline{a} \in \overline{\Sigma}$ and states $p = (p_\varphi)_{\varphi\le\xi}$ and $q = (q_\varphi)_{\varphi\le\xi}$, we have a transition $p \xrightarrow{\overline{a}} q$ in $\mathcal{A}_\xi$ if and only if, for all $\varphi \le \xi$, we have $p_\varphi \xrightarrow{\overline{a}\restriction\varphi} q_\varphi$ in the automaton $\mathcal{B}_{\mathrm{top}(\varphi)}$. Note that a sequence of states $p^0, p^1, \ldots$ defines a run of $\mathcal{A}_\xi$ for a word $\overline{w} \in \overline{\Sigma}^\infty$ if and only if for each $\varphi \le \xi$, its projection $p^0_\varphi, p^1_\varphi, \ldots$ on φ is a run of $\mathcal{B}_{\mathrm{top}(\varphi)}$ for the word $\overline{w}{\restriction}\varphi$. A run of $\mathcal{A}_\xi$ is accepting if and only if for each $\varphi \le \xi$, its projection on $\mathcal{B}_{\mathrm{top}(\varphi)}$ is accepting.

Lemma 5. *Let* $\overline{w} = (w, (w_\varphi)_{\varphi\le\xi}) \in \overline{\Sigma}^\infty$. *Then,* $\overline{w} \in \mathcal{L}(\mathcal{A}_\xi)$ *if and only if for each* $\varphi \le \xi$ *we have* $\mathrm{supp}(w_\varphi) = \varphi^w = \{p < |w| \mid w, p \models \varphi\}$.

Proof. Assume $\overline{w} \in \mathcal{L}(\mathcal{A}_\xi)$. We show that $\varphi^w = \mathrm{supp}(w_\varphi)$ by structural induction on $\varphi \le \xi$. So let $\varphi = M(\varphi_1,\ldots,\varphi_\ell) \le \xi$ such that $\varphi_i^w = \mathrm{supp}(w_{\varphi_i})$ holds for $1 \le i \le \ell$. Since $\overline{w}$ is accepted by the automaton $\mathcal{A}_\xi$, the word

$\overline{w}\restriction\varphi = (w, w_{\varphi_1}, \ldots, w_{\varphi_\ell}, w_\varphi)$ is accepted by $\mathcal{B}_M$. Hence, using Theorem 4 and the hypothesis we get

$$\begin{aligned}\mathrm{supp}(w_\varphi) &= \{p < |w| \mid w \models_{\mathrm{MSO}} [\![M]\!](\mathrm{supp}(w_{\varphi_1}), \ldots, \mathrm{supp}(w_{\varphi_\ell}), p)\} \\ &= \{p < |w| \mid w \models_{\mathrm{MSO}} [\![M]\!](\varphi_1^w, \ldots, \varphi_\ell^w, p)\} \\ &= \varphi^w.\end{aligned}$$

For the other direction, assume that $\varphi^w = \mathrm{supp}(w_\varphi)$ for all $\varphi \leq \xi$. Let $\varphi = M(\varphi_1, \ldots, \varphi_\ell) \leq \xi$. We have $\varphi^w = \{p < |w| \mid w \models_{\mathrm{MSO}} [\![M]\!](\varphi_1^w, \ldots, \varphi_\ell^w, p)\}$ and we get $\mathrm{supp}(w_\varphi) = \{p < |w| \mid w \models_{\mathrm{MSO}} [\![M]\!](\mathrm{supp}(w_{\varphi_1}), \ldots, \mathrm{supp}(w_{\varphi_\ell}), p)\}$ using our hypothesis. Since $\overline{w}\restriction\varphi = (w, w_{\varphi_1}, \ldots, w_{\varphi_\ell}, w_\varphi)$ we deduce from Theorem 4 that $\overline{w}\restriction\varphi$ is accepted by $\mathcal{B}_M$. Since this holds for each $\varphi \leq \xi$ we obtain $\overline{w} \in \mathcal{L}(\mathcal{A}_\xi)$. $\square$

Proposition 6. *The formula ξ is satisfiable if and only if there exists $\overline{w} \in \mathcal{L}(\mathcal{A}_\xi)$ with* $\mathrm{supp}(w_\xi) \neq \emptyset$.

Proof. Assume that ξ is satisfiable. There exist a word $w \in \Sigma^\infty$ and a position p in w with $w, p \models \xi$. For each $\varphi \in \mathrm{TL}(B)$, there is a unique word $w_\varphi \in \{0,1\}^\infty$ with $|w| = |w_\varphi|$ and $\mathrm{supp}(w_\varphi) = \varphi^w$. Let $\overline{w} = (w, (w_\varphi)_{\varphi \leq \xi}) \in \overline{\Sigma}^\infty$. By Lemma 5 we get $\overline{w} \in \mathcal{L}(\mathcal{A}_\xi)$. Moreover, we have $p \in \xi^w = \mathrm{supp}(w_\xi) \neq \emptyset$.

Conversely let $\overline{w} = (w, (w_\varphi)_{\varphi \leq \xi}) \in \mathcal{L}(\mathcal{A}_\xi)$ with $\mathrm{supp}(w_\xi) \neq \emptyset$. By Lemma 5 we get $\emptyset \neq \mathrm{supp}(w_\xi) = \xi^w = \{p < |w| \mid w, p \models \xi\}$. Therefore, ξ is satisfiable. $\square$

Theorem 7. *Let B be a finite set of modality names with associated MSO-modalities. Then the satisfiability problem for* $\mathrm{TL}(B)$ *is in* PSPACE.

Proof. Let ξ be some formula from $\mathrm{TL}(B)$ whose satisfiability we want to check. By Proposition 6, we have to decide whether $\mathcal{A}_\xi$ accepts some word $\overline{w}$ with $\mathrm{supp}(w_\xi) \neq \emptyset$. Recall that a state of $\mathcal{A}_\xi$ is a tuple of states from the automata $\mathcal{B}_M$ whose length is bounded by the size of the formula ξ. Hence a state of $\mathcal{A}_\xi$ requires space polynomial in the size of ξ and the same holds for any letter from $\overline{\Sigma}$. Given two states q and q' of $\mathcal{A}_\xi$ and a letter $a \in \overline{\Sigma}$, one can check in polynomial space whether $q \xrightarrow{a} q'$ in $\mathcal{A}_\xi$. Note that the automata $\mathcal{B}_M$ are fixed and need not be computed. Hence the search for an accepting run can be performed by a nondeterministic Turing machine using space polynomial in the size of ξ. $\square$

A *Kripke structure* is transition system $K = (S, \rightarrow, s, \sigma)$ with S a finite set of states, $\rightarrow \subseteq S^2$ the transition function, $s \in S$ the initial state and $\sigma : S \rightarrow \Sigma$ the labeling function. A formula $\xi \in \mathrm{TL}(B)$ holds in K (written $K \models \xi$) if for all maximal paths $s_0, s_1, \ldots$ in K with $s_0 = s$ we have $\sigma(s_0)\sigma(s_1)\ldots, 0 \models \xi$.

Model checking problem for $\mathrm{TL}(B)$ *over words:* Given a Kripke structure K and a formula $\xi \in \mathrm{TL}(B)$, do we have $K \models \xi$?

Theorem 8. *Let B be a finite set of modality names with associated MSO-modalities. Then the model checking problem for* $\mathrm{TL}(B)$ *is in* PSPACE.

Proof. Let $\xi \in \mathrm{TL}(B)$. The formula $\neg\xi$ is in $\mathrm{TL}(B \cup \{\neg\})$ and we consider the automaton $\mathcal{A}$ obtained from $\mathcal{A}_{\neg\xi}$ by projecting the transition labels to Σ, *i.e.*, $p \xrightarrow{a} q$ in $\mathcal{A}$ if there exists $\overline{a} = (a, (a_\varphi)_{\varphi \leq \neg\xi}) \in \overline{\Sigma}$ with $p \xrightarrow{\overline{a}} q$ in $\mathcal{A}_{\neg\xi}$. Again, a state of $\mathcal{A}$ can be stored in polynomial space and one can check whether $p \xrightarrow{a} q$ in $\mathcal{A}$ in polynomial space. Therefore, applying the usual technique we get a PSPACE algorithm for the model checking problem. $\square$

The actual performance of the algorithms for satisfiability and model checking depend on the basic automata $\mathcal{B}_M$ for $M \in B$. For PLTL, these basic automata have very few states: $\mathcal{B}_a$ for $a \in \Sigma$, $\mathcal{B}_\neg$ and $\mathcal{B}_\vee$ have just one state, $\mathcal{B}_{\mathsf{U}}$ has three states, and all the other automata have two states. Thus, the automaton $\mathcal{A}_\xi$ has at most $2^m \cdot 3^n$ states where m is the number of occurrences of temporal operators different from U and n is the number of occurrences of U in ξ.

5 Local temporal logic over traces

We briefly recall some notions about Mazurkiewicz traces (see [6] for background). A *dependence alphabet* is a pair (Σ, D) where the alphabet Σ is a finite set of actions and the *dependence relation* $D \subseteq \Sigma \times \Sigma$ is reflexive and symmetric.

For a partial order $(V, \leq)$, let $\lessdot$ denote the successor relation $\lessdot \; = \; < \setminus <^2$. Further, $\|$ denotes incomparability, *i.e.*, $\| = V^2 \setminus (\leq \cup \geq)$. A *(Mazurkiewicz) trace* is a finite or infinite labeled partial order $t = (V, \leq, \lambda)$ where V is a set of vertices labeled by $\lambda : V \to \Sigma$ and $\leq$ is a partial order relation on V satisfying the following conditions:

1. for all $y \in V$, the set $\downarrow y = \{x \in V \mid x \leq y\}$ is finite,
2. $x \parallel y$ implies $(\lambda(x), \lambda(y)) \notin D$ for all $x, y \in V$, and
3. $x \lessdot y$ implies $(\lambda(x), \lambda(y)) \in D$ for all $x, y \in V$.

The set of all traces is denoted $\mathbb{R}(\Sigma, D)$.

We now interpret monadic second order formulas over traces. The semantics for traces is defined as for words in Section 2. Let $t = (V, \leq, \lambda)$ be a trace. A *valuation in* t for the formula φ is now a mapping ν that assigns elements of V to free individual variables of φ and subsets of V to free set variables of φ. The definition of satisfaction $t, \nu \models_{\mathrm{MSO}} \varphi$ can be taken verbatim from Section 2 with the only exception that $t, \nu \models_{\mathrm{MSO}} P_a(x)$ if and only if $\lambda(\nu(x)) = a$. It should be noted that $\nu(x) \leq \nu(y)$ refers now to the partial order of the trace.

Similarly, the temporal logic $\mathrm{TL}(B)$ is defined as in Definition 2. The only difference is that the semantics φ^t is now defined for a trace t:

$$\varphi^t = \{p \in V \mid t \models_{\mathrm{MSO}} [\![M]\!](\varphi_1^t, \dots, \varphi_\ell^t, p)\}$$

and as before, we write $t, p \models \varphi$ for $p \in \varphi^t$.

In the next section we show that the satisfiability problem and the model checking problem are decidable in PSPACE for $\mathrm{TL}(B)$ when B is finite. But first, we show that all modalities that were considered so far in local logics for traces can be defined in our setting. As a corollary, we get that all local temporal logics for traces considered so far are decidable in PSPACE.

We start with event based temporal logics and will consider later process based ones. In addition to the constants Σ and the boolean connectives $\neg$ and $\vee$, these logics are build using various temporal modalities described below.

Universal until. The simplest logic $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$ studied in [4] uses only two modalities EX of arity 1 and U of arity 2 (there are some technical subtleties about initial modalities or initial satisfiability of a formula that will be discussed later). Intuitively, $\mathsf{EX}\,\varphi$ means that there is an immediate successor of the current vertex where φ holds. The universal until $\varphi\,\mathsf{U}\,\psi$ claims the existence of a vertex z in the future of the current one x such that ψ holds at z and φ holds for all vertices between x and z. Formally, we have $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U}) = \mathrm{TL}(\Sigma \cup \{\neg, \vee, \mathsf{EX}, \mathsf{U}\})$ if EX and U are defined by the following MSO-modalities.

$$\begin{aligned}
[\![\mathsf{EX}]\!](X_1, x) &= \exists z(x < z \wedge X_1(z) \wedge \forall y(x < y \leq z \rightarrow y = z)) \\
[\![\mathsf{U}]\!](X_1, X_2, x) &= \exists z(x \leq z \wedge X_2(z) \wedge \forall y(x \leq y < z \rightarrow X_1(y)))
\end{aligned}$$

The logic $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U})$ is expressively complete with respect to $\mathrm{FO}_\Sigma(<)$, the first order theory of traces if and only if the dependence alphabet is a cograph [4]. The satisfiability problem was shown to be PSPACE-complete. The hardness follows from the corresponding result on words. The PSPACE algorithm is obtained using alternating automata. Though not all details were given, the proof of this upper bound was more than 4 pages long in [5]. Since $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{U}) = \mathrm{TL}(\Sigma \cup \{\neg, \vee, \mathsf{EX}, \mathsf{U}\})$, it is a trivial corollary of Theorem 9.

Filtered until. In order to obtain expressive completeness for arbitrary dependence alphabets, [7] considered $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{EY}, \mathsf{U}_C, \mathsf{S}_C)$ where $C \subseteq \Sigma$. Compared to the universal until U, the filtered universal until U_C adds an alphabetic requirement on the vertices that are below z but not below x. The modalities EY and S_C are the past versions of EX and U_C. We can express this logic in our framework, $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{EY}, \mathsf{U}_C, \mathsf{S}_C) = \mathrm{TL}(\Sigma \cup \{\neg, \vee, \mathsf{EX}, \mathsf{EY}, \mathsf{U}_C, \mathsf{S}_C\})$ if we associate with EY, U_C and S_C the following MSO-modalities.

$$\begin{aligned}
[\![\mathsf{EY}]\!](X_1, x) &= \exists z(z < x \wedge X_1(z) \wedge \forall y(z \leq y < x \rightarrow y = z)) \\
[\![\mathsf{U}_C]\!](X_1, X_2, x) &= \exists z(x \leq z \wedge X_2(z) \wedge \forall y(x \leq y < z \rightarrow X_1(y)) \\
&\qquad \wedge \forall y(y \leq z \wedge \textstyle\bigvee_{c \in C} P_c(y) \rightarrow y \leq x)) \\
[\![\mathsf{S}_C]\!](X_1, X_2, x) &= \exists z(z \leq x \wedge X_2(z) \wedge \forall y(z < y \leq x \rightarrow X_1(y)) \\
&\qquad \wedge \forall y(y \leq x \wedge \textstyle\bigvee_{c \in C} P_c(y) \rightarrow y \leq z))
\end{aligned}$$

In [7], the logic $\mathrm{LocTL}_\Sigma(\mathsf{EX}, \mathsf{EY}, \mathsf{U}_C, \mathsf{S}_C)$ was shown to be expressively complete with respect to $\mathrm{FO}_\Sigma(<)$ for arbitrary dependence alphabets. The satisfiability problem was also shown to be decidable in PSPACE using two-way alternating automata, the proof being long and non trivial. Again this complexity upper bound becomes a trivial corollary of Theorem 9.

We say that $\mathsf{EX}, \mathsf{EY}, \mathsf{U}_C$ and S_C are *first order modalities* because $[\![\mathsf{EX}]\!]$, $[\![\mathsf{EY}]\!]$, $[\![\mathsf{U}_C]\!]$ and $[\![\mathsf{S}_C]\!]$ use quantification over individual variables only. The temporal logics defined with FO-modalities are thus trivially contained in $\mathrm{FO}_\Sigma(<)$. We will see now a temporal logic using some modalities that are not FO-definable.

Existential until. The temporal logic for causality (TLC) was introduced in [1]. In our framework, it can be defined by $\mathrm{TL}(\Sigma \cup \{\neg, \vee, \mathsf{EX}, \mathsf{EY}, \mathsf{Eco}, \mathsf{EG}, \mathsf{EU}, \mathsf{ES}\})$. Intuitively, $\mathsf{Eco}\,\varphi$ claims that φ holds for some vertex concurrent to the current one. The formula $\varphi\,\mathsf{EU}\,\psi$ holds if there is a path starting at the current vertex in the Hasse diagram of the trace such that φ holds along the path until ψ holds. Similarly, $\mathsf{EG}\,\varphi$ claims the existence of a maximal path in the Hasse diagram of the trace, starting from the current vertex, where φ always holds. Finally, ES is the past version of EU. Formally, the semantics of TLC is obtained with the following MSO-modalities.

$$
\begin{aligned}
[\![\mathsf{Eco}]\!](X_1, x) &= \exists z(\neg(x \le z) \wedge \neg(z \le x) \wedge X_1(z)) \\
[\![\mathsf{EU}]\!](X_1, X_2, x) &= \exists z(x \le z \wedge X_2(z) \wedge \exists Y(\forall y(Y(y) \wedge y < z \rightarrow X_1(y)) \wedge \\
&\qquad Y \text{ is a maximal totally ordered set contained in } {\uparrow}x \cap {\downarrow}z)) \\
[\![\mathsf{ES}]\!](X_1, X_2, x) &= \exists z(z \le x \wedge X_2(z) \wedge \exists Y(\forall y(Y(y) \wedge z < y \rightarrow X_1(y)) \wedge \\
&\qquad Y \text{ is a maximal totally ordered set contained in } {\downarrow}x \cap {\uparrow}z)) \\
[\![\mathsf{EG}]\!](X_1, x) &= \exists Y(\forall y(Y(y) \rightarrow X_1(y)) \wedge \\
&\qquad Y \text{ is a maximal totally ordered set contained in } {\uparrow}x)
\end{aligned}
$$

TLC was proved to be decidable in PSPACE in [1] using a tableau construction. Again, this upper bound becomes a corollary of Theorem 9. The expressiveness results for TLC were established in [4]. For cograph dependence alphabets TLC has the same expressive power as $\mathrm{FO}_\Sigma(<)$, but due to the claim of the existence of a path in the modality EU it is not contained in FO for arbitrary dependence alphabets.

Initial satisfiability. A given formula $\xi \in \mathrm{TL}(B)$ is satisfiable over traces if there exists a trace $t \in \mathbb{R}(\Sigma, D)$ and some position p in t such that $t, p \models \xi$. Since a trace does not necessarily have a unique minimal position, there is no canonical way to define *initial* satisfiability over traces. Two approaches have been considered.

In [4], an initial modality $\mathsf{EM}\,\varphi$ was introduced with the meaning $t \models \mathsf{EM}\,\varphi$ if there is a minimal position p in t with $t, p \models \varphi$. Then, an initial formula α is a boolean combination of initial modalities and the initial satisfiability problem is to know whether there exists a trace $t \in \mathbb{R}(\Sigma, D)$ with $t \models \alpha$. To cope with this approach, we associate with EM the MSO modality

$$[\![\mathsf{EM}]\!](X_1, x) = \exists y(X_1(y) \wedge \neg\exists z(z < y)).$$

Then, the formula $\alpha \in \mathrm{LocTL}_\Sigma(\cdots)$ is initially satisfiable over traces if and only if the formula $\alpha \in \mathrm{TL}(B)$ is satisfiable (with $[\![-]\!]$) over traces.

In [1] a dual approach is taken which can be dealt with in the same way. Here, it is said that a a local formula φ is initially satisfiable if there exists a trace t such that φ holds at all minimal vertices of t, i.e., $t \models \neg\,\mathsf{EM}\,\neg\varphi$.

The other approach used in [3] is to consider rooted traces. Let $\# \notin \Sigma$ and $t = (V, \leq, \lambda) \in \mathbb{R}(\Sigma, D)$. The rooted trace associated with t is $\# \cdot t = (V \cup \{\#\}, \leq \cup (\{\#\} \times (V \cup \{\#\})), \lambda \cup (\# \mapsto \#)$. It is a trace over the alphabet $\Sigma' = \Sigma \cup \{\#\}$ and the dependence relation $D' = D \cup (\{\#\} \times \Sigma) \cup (\Sigma \times \{\#\})$. Then, we say that a local formula $\varphi \in \mathrm{LocTL}_\Sigma(\cdots)$ is initially satisfiable if there exists a trace $t \in \mathbb{R}(\Sigma, D)$ such that $\# \cdot t, \# \models \varphi$. To cope with this approach, we add a modality name init of arity 1 to B with associated MSO-modality

$$[\![\mathrm{init}]\!](X_1, x) = \exists y(X_1(y) \wedge P_\#(y) \wedge \forall z(y \leq z) \wedge \forall z(P_\#(z) \to z = y)).$$

Then, the formula $\varphi \in \mathrm{LocTL}_\Sigma(\cdots)$ is initially satisfiable over $\mathbb{R}(\Sigma, D)$ if and only if the formula $\mathrm{init}(\varphi) \in \mathrm{TL}(B)$ is satisfiable (with $[\![-]\!]$) over $\mathbb{R}(\Sigma', D')$.

Process-based modalities. We conclude the section by showing that the temporal logic over traces TrPTL introduced by Thiagarajan [13] can also be dealt with in our framework. The underlying idea is that the actions of the dependence alphabet are executed by independent processes. Communication between these processes is possible by the execution of joint actions. Hence, with any action $a \in \Sigma$, we associate a nonempty set of processes $p(a) \subseteq \{1, 2, \ldots, n\}$ in such a way that $(a, b) \in D$ iff $p(a) \cap p(b) \neq \emptyset$. This ensures that events performed by process i are linearly ordered in any trace t. With this additional information, one can define modalities that speak about the location of an action. The logic TrPTL is based on modalities p_i, $\mathcal{O}_i$ and U_i ($i \in \{1, \ldots, n\}$) of arity 0, 1 and 2 respectively.

The semantics given in [13] is that of a global temporal logic. Hence it may come as a surprise that we can deal with it in our framework. But actually, apart initially, formulas are evaluated at *prime* configurations, i.e., configurations having exactly one maximal element. By identifying a prime configuration with its maximal vertex we see that the logic is actually local. Intuitively, p_i holds if the current vertex is located on process i and $\mathcal{O}_i\varphi$ means that φ holds at the first vertex of process i which is not below the current one. Finally, $\varphi \mathsf{U}_i \psi$ means that we have φ until ψ on the sequence of vertices located on process i and starting from the last vertex of process i which is below the current one. Formally, the semantics is defined as follows using the macro $P_i(x) = \bigvee_{\{c \mid i \in p(c)\}} P_c(x)$:

$$\begin{aligned}
[\![p_i]\!](x) &= P_i(x) \\
[\![\mathcal{O}_i]\!](X_1, x) &= \exists y(X_1(y) \wedge P_i(y) \wedge \neg(y \leq x) \wedge \forall z(P_i(z) \to (z \leq x \vee y \leq z))) \\
[\![\mathsf{U}_i]\!](X_1, X_2, x) &= \exists y(P_i(y) \wedge y \leq x \wedge \forall z(P_i(z) \wedge z \leq x \to z \leq y) \\
&\qquad\qquad \wedge \exists z(P_i(z) \wedge y \leq z \wedge X_2(z) \\
&\qquad\qquad\qquad \wedge \forall u((P_i(u) \wedge y \leq u < z) \to X_1(u))))
\end{aligned}$$

TrPTL was proved to be decidable in EXPTIME in [13] using a difficult result on *gossip* automata over traces [10]. As a corollary of Theorem 9, we can improve this upper bound to PSPACE. Since the logic TrPTL is defined by FO-modalities, it is contained in $\mathrm{FO}_\Sigma(<)$ but the precise expressive power of TrPTL is still unknown.

6 Complexity of local temporal logics for traces

We want to show that the following problem is decidable in PSPACE.

Satisfiability problem for TL(B) *over traces:* Given a formula $\xi \in \mathrm{TL}(B)$, does there exist a trace $t \in \mathbb{R}(\Sigma, D)$ and some position p in t such that $t, p \models \xi$?

This will be done by a reduction to Theorem 7. For this reason, we first recall the relation between words and traces, more details can be found in [6].

Let $t = (V, \leq, \lambda)$ be a trace and let $\sqsubseteq$ be any linear extension of $\leq$ of order type at most ω. Then we can view $(V, \sqsubseteq, \lambda)$ as a word $w \in \Sigma^\infty$. The set of *linearizations* $\mathrm{Lin}(t) \subseteq \Sigma^\infty$ of t is the set of all words $w \in \Sigma^\infty$ that arise in this way. Conversely, each word $w \in \Sigma^\infty$ is the linearization of a unique trace $t \in \mathbb{R}(\Sigma, D)$.

In the following, we will evaluate MSO formulas over words and over traces. To make this clear, we use $\models^t_{\mathrm{MSO}}$ for traces and $\models^w_{\mathrm{MSO}}$ for words (though the context is sufficient to distinguish between the two). There exists a FO formula $\eta(x, y)$ with two free individual variables such that for all traces $t \in \mathbb{R}(\Sigma, D)$, words $w \in \mathrm{Lin}(t)$ and vertices $p, q \in V$, we have $t \models^t_{\mathrm{MSO}} p \leq q$ if and only if $w \models^w_{\mathrm{MSO}} \eta(p, q)$. Let φ be an MSO formula. We denote by $\overline{\varphi}$ the MSO formula obtained by replacing in φ any subformula of the form $x \leq y$ by $\eta(x, y)$. Then, we have for all traces $t \in \mathbb{R}(\Sigma, D)$, words $w \in \mathrm{Lin}(t)$ and valuations ν in V, $t, \nu \models^t_{\mathrm{MSO}} \varphi$ if and only if $w, \nu \models^w_{\mathrm{MSO}} \overline{\varphi}$.

After these preliminary remarks, fix some set B of modality names together with their arity function and associated MSO-modality defined by the mapping $[\![-]\!] : B \to \mathrm{MSO}_\Sigma(<)$. This defines a temporal logic TL(B) whose interpretation over traces with $[\![-]\!]$ is denoted $\models^t_{[\![-]\!]}$. We also consider the mapping $\overline{[\![-]\!]} : B \to \mathrm{MSO}_\Sigma(<)$ so that for $M \in B$, $\overline{[\![M]\!]}$ is obtained by replacing in $[\![M]\!]$ any subformula of the form $x \leq y$ by $\eta(x, y)$. The interpretation of TL(B) over words with $\overline{[\![-]\!]}$ is denoted $\models^w_{\overline{[\![-]\!]}}$. We obtain the following essential link between the two semantics: for all $\xi \in \mathrm{TL}(B)$, for all traces $t \in \mathbb{R}(\Sigma, D)$, all words $w \in \mathrm{Lin}(t)$ and all positions p in t, we have $t, p \models^t_{[\![-]\!]} \xi$ if and only if $w, p \models^w_{\overline{[\![-]\!]}} \xi$.

Therefore, the formula ξ is satisfiable over traces with the MSO-modalities $[\![-]\!]$ if and only if it is satisfiable over words with the MSO-modalities $\overline{[\![-]\!]}$. Since, by Theorem 7, this latter question is decidable in space polynomial in the size of ξ, we obtain the following

Theorem 9. *Let* (Σ, D) *be a dependence alphabet,* B *a finite set of modality names with associated MSO-modalities. Then the satisfiability problem for* TL(B) *over traces is decidable in PSPACE.*

We turn now to the model checking problem. In order to give its definition, we first introduce *asynchronous Kripke structures.* We need to fix some notation. Let Loc be a finite set of locations and let Q_i be a finite set for each $i \in \mathrm{Loc}$. We let $Q_I = \prod_{i \in I} Q_i$ for $I \subseteq \mathrm{Loc}$ and if $q = (q_i)_{i \in \mathrm{Loc}} \in Q_{\mathrm{Loc}}$ then we let $q_I = (q_i)_{i \in I}$ for $I \subseteq \mathrm{Loc}$. An *asynchronous Kripke structure* (AKS for short) is a tuple $\mathrm{AK} = ((Q_i)_{i\in\mathrm{Loc}}, (\delta_I)_{I\subseteq\mathrm{Loc}}, q^0, (\sigma_i)_{i\in\mathrm{Loc}})$ where Q_i is a finite set of local states for process i, $\delta_I \subseteq Q_I \times Q_I$ is a local transition relation, $q^0 \in Q_{\mathrm{Loc}}$ is the global initial state, and $\sigma_i : Q_i \to 2^{\mathrm{AP}_i}$ assigns to each local state the set of atomic propositions from the finite set AP_i that holds in this states.

A run of AK is (an isomorphism class of) a labelled partial order $\rho = (V, \leq, \ell, W)$ where a vertex $v \in V$ represents the occurrence of a transition, $\leq$ is the ordering between transitions, $\ell : V \to 2^{\mathrm{Loc}} \setminus \{\emptyset\}$ gives for each transition v the nonempty set $\ell(v)$ of processes taking part in it and W assigns to each transition $v \in V$ the tuple $W(v) \in Q_{\ell(v)}$ of updated states for the processes in $\ell(v)$. We require that

1. for all $v \in V$, the set $\downarrow v = \{u \in V \mid u \leq v\}$ is finite,
2. $u \parallel v$ implies $\ell(u) \cap \ell(v) = \emptyset$ for all $u, v \in V$, and
3. $u \lessdot v$ implies $\ell(u) \cap \ell(v) \neq \emptyset$ for all $u, v \in V$.

This implies in particular that two transitions cannot read or write simultaneously the same process. Finally, the transition relations of AK must be satisfied: for $v \in V$, let $R(v) = (R_i(v))_{i\in\ell(v)}$ be defined by $R_i(v) = q_i^0$ if $\{u < v \mid i \in \ell(u)\} = \emptyset$ and $R_i(v) = W_i(\max(\{u < v \mid i \in \ell(u)\}))$ otherwise. Then, we must have $(R(v), W(v)) \in \delta_{\ell(v)}$ for all $v \in V$.

If $\rho = (V, \leq, \ell, W)$ is a run of AK and $U \subseteq V$ is such that $U = \downarrow U = \{v \in V \mid v \leq u \text{ for some } u \in U\}$ then the restriction $(U, \leq, \ell, W)$ of ρ to U is also a run of AK which is called a prefix of ρ. A run of AK is maximal if it is not a strict prefix of some other run of AK.

Without loss of generality, we may assume that $\sigma_i(q_i) \neq \emptyset$ for all $q_i \in Q_i$ and that the sets AP_i are pairwise disjoint. Let $\mathrm{AP} = \biguplus_{i\in\mathrm{Loc}} \mathrm{AP}_i$ and $\Sigma = 2^{\mathrm{AP}} \setminus \{\emptyset\}$. For $a \in \Sigma$ we let $\mathrm{loc}(a) = \{i \in \mathrm{Loc} \mid \mathrm{AP}_i \cap a \neq \emptyset\}$. The dependence relation over Σ is defined by $(a, b) \in D$ if $\mathrm{loc}(a) \cap \mathrm{loc}(b) \neq \emptyset$. With each run $\rho = (V, \leq, \ell, W)$ of AK we associate $\tau(\rho) = (V, \leq, \lambda)$ where $\lambda(v) = \bigcup_{i\in\ell(v)} \sigma_i(W_i(v))$. It is not hard to see that $\tau(\rho)$ is a trace over (Σ, D).

An asynchronous Kripke structure AK satisfies a temporal formula $\xi \in \mathrm{TL}(B)$ ($\mathrm{AK} \models \xi$) if, for any maximal run ρ of AK, we have $\# \cdot \tau(\rho), \# \models \xi$.

Model checking problem for $\mathrm{TL}(B)$ *and AKS:* Given an asynchronous Kripke structure AK and a formula $\xi \in \mathrm{TL}(B)$, do we have $\mathrm{AK} \models \xi$?

Theorem 10. *Let* $(\mathrm{AP}_i)_{i\in\mathrm{Loc}}$ *and* (Σ, D) *be as above. Let* B *be a finite set of modality names with associated MSO-modalities over the alphabet* Σ*. Then the model checking problem for* $\mathrm{TL}(B)$ *and AKS is decidable in PSPACE.*

Proof. Let $\mathrm{AK} = ((Q_i)_{i\in\mathrm{Loc}}, (\delta_I)_{I\subseteq\mathrm{Loc}}, q^0, (\sigma_i)_{i\in\mathrm{Loc}})$ be an AKS. We define an associated sequential (global) Kripke structure $K = (S, \delta, s_0, \sigma)$. The set of global states is $S = Q_{\mathrm{Loc}} \times 2^{\mathrm{Loc}}$ and $s_0 = (q^0, \mathrm{Loc})$ is the initial global state. The transition relation $\delta \subseteq S \times S$ is defined by $((p, I), (q, J)) \in \delta$ if $J \neq \emptyset$, $(p_J, q_J) \in \delta_J$ and $p_{\overline{J}} = q_{\overline{J}}$ where $\overline{J} = \mathrm{Loc} \setminus J$. Finally, the labelling $\sigma : S \to \Sigma$ is given by $\sigma(q, I) = \bigcup_{i\in I} \sigma_i(q_i)$.

Runs of K correspond to linearizations of runs of AK. More precisely, let $\rho = (V, \leq, \ell, W)$ be a run of AK and let $\sqsubseteq$ be any linear extension of $\leq$ of order type at most ω. We can write $V = \{v_1, v_2, \ldots\}$ with $v_{n-1} \sqsubseteq v_n$. We define a sequence of global states $s_n = (q^n, I_n)$ by $I_0 = \mathrm{Loc}$ and for $n > 0$, $I_n = \ell(v_n)$, $q^n_{I_n} = W(v_n)$ and $q^n_{\overline{I_n}} = q^{n-1}_{\overline{I_n}}$. Then, $s_0 s_1 \cdots$ is a run of K which is a linearization of ρ. Moreover, the word $\sigma(s_0)\sigma(s_1)\ldots \in \Sigma^\infty$ is a linearization of the trace $\tau(\rho)$. Conversely, any run of K is a linearization of some run of AK.

For the model checking problem, we are interested in maximal runs. Clearly, a linearization of a maximal run of AK is a maximal run of K. Conversely, a maximal *finite* run of K is a linearization of a maximal finite run of AK. Now, an infinite run $(q^0, \mathrm{Loc})(q^1, I_1)(q^2, I_2)\ldots$ of K is a linearization of a maximal run of AK if and only if eventually, there is no enabled transition involving a set of processes that participate in finitely many transitions of the run: there exists $N \geq 0$ such that for all $\emptyset \neq J \subseteq \mathrm{Loc}$ with $J \cap I_n = \emptyset$ for all $n > N$, we have $(\{q^N_J\} \times Q_J) \cap \delta_J = \emptyset$. We call a run of K accepting if it is either finite and maximal or infinite and satisfies the above condition (which by the way can be described with a Muller table). Hence, accepting runs of K correspond to maximal runs of AK.

Now, let $\xi \in \mathrm{TL}(B)$. We use the notation introduced for the satisfiability. Then AK $\models^t_{\llbracket - \rrbracket} \xi$ if and only if for all accepting runs $s_0 s_1 \ldots$ of K we have $\sigma(s_0)\sigma(s_1)\ldots, 0 \models^w_{\overline{\llbracket - \rrbracket}} \xi$. Therefore, we are reduced to a model checking problem of a Kripke structure K with some acceptance condition on infinite runs.

Note that a state of K can be stored in space polynomial in the size of AK. Also, the same space bound suffices to decide whether a pair of states (s, s') forms a transition of K and to compute $\sigma(s)$. Finally searching for a loop that satisfies the acceptance condition can also be done in space polynomial in the size of AK. One just has to guess at the beginning of the loop the set J of processes that will not participate in the transitions of the loop. This guess is easy to check within the polynomial space bound as well as the fact that no transition involving a set of processes contained in J is enabled at the beginning of the loop. Therefore, using for ξ (interpreted with $\overline{\llbracket - \rrbracket}$) the technique described in the proof of Theorem 8, a slight modification of the usual model checking procedure allows to solve our problem in PSPACE. $\square$

The theorems above show that for any of the local temporal logics introduced in Section 5, the satisfiability and the model checking problems become decidable in PSPACE. For some of these logics, this result was known, for TrPTL [13], it seems to be new.

7 Generalizations

The framework of MSO-definable local temporal logics extends verbatim to more general partial orders than Mazurkiewicz traces. The difficulty is to find reasonable classes of partial orders such that complexity results can be obtained for the satisfiability and the model checking problems. For instance, we can show that for the class of all Message sequence charts (MSCs), the satisfiability for a very restricted local temporal logic (namely, a small fragment of TLC^-) is undecidable. On the other hand, there are natural subclasses of MSCs for which the satisfiability problem is decidable in PSPACE. These results will appear in a forthcoming paper.

References

1. R. Alur, R. Peled, and W. Penczek. Model checking of causality properties. In *LICS 95*, pages 90–100. IEEE Computer Society Press, 1995.
2. J.R. Büchi. On a decision method in restricted second order arithmetics. In E. Nagel et al., editors, *Proc. Intern. Congress on Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford University Press, Stanford, 1960.
3. V. Diekert. A pure future local temporal logic beyond cograph-monoids. In M. Ito, editor, *Proc. of the RIMS Symposium on Algebraic Systems, Formal Languages and Conventional and Unconventional Computation Theory, Kyoto, Japan 2002*, 2002.
4. V. Diekert and P. Gastin. Local temporal logic is expressively complete for cograph dependence alphabets. In *LPAR 01*, Lecture Notes in Artificial Intelligence vol. 2250, pages 55–69. Springer, 2001.
5. V. Diekert and P. Gastin. Local temporal logic is expressively complete for cograph dependence alphabets. Tech. Rep. LIAFA, Université Paris 7 (France), 2003. http://www.liafa.jussieu.fr/∼gastin/Articles/diegas03.html.
6. V. Diekert and G. Rozenberg. *The Book of Traces.* World Scientific Publ. Co., 1995.
7. P. Gastin and M. Mukund. An elementary expressively complete temporal logic for Mazurkiewicz traces. In *Proc. of ICALP'02*, number 2380 in LNCS, pages 938–949. Springer Verlag, 2002.
8. H.W. Kamp. *Tense logic and the theory of linear order.* PhD thesis, University of California, Los Angeles, USA, 1968.
9. O. Kupferman, N. Piterman, and M.Y. Vardi. Extended temporal logic revisited. In *Proc. of CONCUR'01*, number 2154 in LNCS, pages 519–535. Springer Verlag, 2001.
10. M. Mukund and M. Sohoni. Keeping trace of the latest gossip: bounded timestamps suffice. In *Proc. of FST&TCS'93*, number 761 in LNCS, pages 388–399. Springer Verlag, 1993.
11. M. Mukund and P.S. Thiagarajan. Linear time temporal logics over Mazurkiewicz traces. In *Proc. of MFCS'96*, number 1113 in LNCS, pages 62–92. Springer Verlag, 1996.
12. A. Rabinovich and S. Maoz. An infinite hierarchy of temporal logics over branching time. *Information and Computation*, 171(2):306–332, 2001.
13. P.S. Thiagarajan. A trace based extension of linear time temporal logic. In *Proc. of LICS'94*, pages 438–447. IEEE Computer Society Press, 1994.
14. P.S. Thiagarajan. A trace consistent subset of PTL. In *Proc. of CONCUR'95*, number 962 in LNCS, pages 438–452, 1995.
15. W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 133–191. Elsevier Science Publ. B.V., 1990.
16. M.Y. Vardi and P. Wolper. Reasonning about infinite computations. *Information and Computation*, 115:1–37, 1994.
17. P. Wolper. Temporal logic can be more expressive. *Inf. and Control*, 56:72–99, 1983.