Uniform satisfiability problem for local temporal logics over Mazurkiewicz traces*

Paul Gastin[1] and Dietrich Kuske[2]

[1] LSV, CNRS & ENS de Cachan
61, Av. du Président Wilson, F-94235 Cachan Cedex, France,
Paul.Gastin@lsv.ens-cachan.fr
[2] Institut für Informatik, Universität Leipzig
Augustusplatz 10-11, D-04109 Leipzig, Germany,
kuske@informatik.uni-leipzig.de

Abstract. We continue our study of the complexity of temporal logics over concurrent systems that can be described by Mazurkiewicz traces. In a previous paper (CONCUR 2003), we investigated the class of local and MSO definable temporal logics that capture all known temporal logics and we showed that the satisfiability problem for any such logic is in PSPACE (provided the dependence alphabet is fixed). In this paper, we concentrate on the uniform satisfiability problem: we consider the dependence alphabet (i.e., the architecture of the distributed system) as part of the input. We prove lower and upper bounds for the uniform satisfiability problem that depend on the number of monadic quantifier alternations present in the chosen MSO-modalities.

1 Introduction

Executions of distributed systems can be modeled as Mazurkiewicz traces [5] where the architecture of the system is mirrored by the dependence alphabet. Then a trace is a partial order execution of such a system. 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 (cf. [13, 14, 11, 2, 1, 9, 10, 3, 4]). 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 particular local temporal logics are of interest here due to their good algorithmic properties. The common feature of these logics is that formulas are evaluated at single events corresponding to local views of processes. In [8], we proposed a unified treatment of all these local temporal logics very much in the spirit of [7]. Basically, a local temporal logic is given by a finite set of modality names. The semantics of any such modality name is described by a monadic second order (MSO) formula

* Work partly supported by the DAAD-PROCOPE project Temporal and Quantitative Analysis of Distributed Systems.

having a single individual free variable. For any fixed dependence alphabet (i.e., architecture of a distributed system) we showed that the satisfiability problem of any such logic is in PSPACE. For (almost) all temporal logics considered in the literature so far, this was known before. Our contribution was a uniform proof that would also be applicable for not-yet-defined temporal logics.

A more realistic setting is the uniform satisfiability problem where both, the temporal formula and the architecture form the input. In other words, this uniform satisfiability problem for the local temporal logic TL asks whether a given property $\varphi \in$TL can be satisfied in a given architecture (Σ, D) (described as a trace alphabet). The paper at hand studies the complexity of this problem depending on the temporal logic TL. Recall that the semantics of the modality names of TL are given by MSO formulas. The complexity of the uniform satisfiability problem depends on the number of alternations of set quantifiers in these formulas. The bad news is that any quantifier alternation in the MSO-descriptions of the modalities adds an exponent to the space complexity. The good news is that local temporal logics considered in the literature do not have any alternation of set quantifiers and are, more precisely, definable in $\mathrm{M}\Delta_1^1$.

Section 2 defines and discusses the necessary concepts used in this paper. The following Section 3 proves an upper bound on the complexity of the uniform satisfiability problem. Our decision procedure makes crucial use of a locality theorem due to Schwentick & Bartelmann [12] that generalizes both, Hanf's and Gaifman's locality theorems. Section 4 presents, for any $n \in \mathbb{N}$, a local temporal logic whose uniform satisfiability is hard for n-fold exponential space. Examples of temporal logics that fall into our setting can be found in Sections 2 (where we discuss action based logics) and 5 that is devoted to process based logics, e.g., Thiagarajan's logic TrPTL from [13]. From our upper bound, it follows that any temporal logic from the literature has a uniform satisfiability problem whose space complexity is doubly exponential in the alphabet and polynomial in the formula.

Due to space limitations, we had to omit complete proofs as well as many more examples. They can be found in a technical report available on the web pages of the authors.

2 Preliminaries

Throughout this paper, we fix some countably infinite set N of *action names.* A *dependence alphabet* is a pair (Σ, D) where $\Sigma \subset \mathrm{N}$ is a set of action names and $D \subseteq \Sigma^2$ is a symmetric and reflexive relation on Σ. A *trace over* (Σ, D) is a labeled at most countably infinite partial order $(V, \leq, \lambda)$ such that $(V, \leq)$ is a partial order and $\lambda : V \to \Sigma$ is the labeling function satisfying for all $x, y \in V$

- $\downarrow x = \{z \in V \mid z \leq x\}$ is finite
- $(\lambda(x), \lambda(y)) \in D$ implies $x \leq y$ or $y \leq x$
- $x \lessdot y$ implies $(\lambda(x), \lambda(y)) \in D$,

where $\lessdot = <\setminus<^2$ is the immediate successor relation. The set $\mathbb{M}(\Sigma, D)$ comprises all finite traces while $\mathbb{R}(\Sigma, D)$ contains all traces over (Σ, D).

Trace concatenation is an operation $\cdot : \mathbb{M}(\Sigma, D) \times \mathbb{R}(\Sigma, D) \to \mathbb{R}(\Sigma, D)$ defined by $(V, \leq, \lambda) \cdot (V', \leq', \lambda') = (V \uplus V', (\leq \cup \leq' \cup E)^*, \lambda \cup \lambda')$ with $E = \{(v, v') \in V \times V' \mid (\lambda(v), \lambda'(v')) \in D\}$. Its restriction to finite traces is associative, i.e., $(\mathbb{M}(\Sigma, D), \cdot)$ is a monoid, called *trace monoid*.

We can identify a letter $a \in \Sigma$ with the trace $[a] = (\{0\}, \leq, \lambda)$ with $\lambda(0) = a$. In this sense, the trace monoid $\mathbb{M}(\Sigma, D)$ is generated by the set of letters $a \in \Sigma$. The canonical homomorphism $[.] : \Sigma^* \to \mathbb{M}(\Sigma, D)$ can be extended naturally to infinite words: for a (finite or infinite) word $u = a_0 a_1 \ldots$ with $a_i \in \Sigma$, the trace $[u] = (V, \sqsubseteq, \lambda)$ is given by $V = \{i \in \mathbb{N} \mid 0 \leq i < |u|\}$, $\sqsubseteq = E^*$ with $(i, j) \in E$ iff $i < j$ and $(a_i, a_j) \in D$, and $\lambda(i) = a_i$.

Formulas of the logic $\mathrm{MSO}(\mathrm{N}, \lessdot, \mathrm{fin})$ will be interpreted over traces. This logic is based on atomic propositions of the form $(\lambda(x) = a)$ for $a \in \mathrm{N}$, $x \lessdot y$, $x = y$, $x \in X$, and $\mathrm{fin}(X)$ for x, y individual variables and X a set variable. Intuitively, the formula $\mathrm{fin}(X)$ means that the set X is finite. Note that we do not allow the partial order $\leq$ to be used in our formulas. On one hand, the successor relation is sufficient since the partial order can be expressed using the successor. On the other hand, our upper bound proof relies on the fact that the Hasse diagram of any trace has bounded degree. The fragment $\mathrm{MSO}(\mathrm{N}, \lessdot)$ of $\mathrm{MSO}(\mathrm{N}, \lessdot, \mathrm{fin})$ consists of all formulas that do not mention the atomic proposition $\mathrm{fin}(X)$.

Example 2.1. Consider the following two formulas

$$\begin{aligned} \mathrm{upset}(x, X) &= \forall y (y \in X \leftrightarrow y = x \vee \exists z (z \in X \wedge z \lessdot y)) \text{ and} \\ \mathrm{downset}(x, X) &= \mathrm{fin}(X) \wedge \forall y (y \in X \leftrightarrow y = x \vee \exists z (z \in X \wedge y \lessdot z)) \ . \end{aligned}$$

of $\mathrm{MSO}(\mathrm{N}, \lessdot)$ and $\mathrm{MSO}(\mathrm{N}, \lessdot, \mathrm{fin})$, respectively. Then, for a trace $t = (V, \leq, \lambda)$ (over any finite dependence alphabet), $t \models \mathrm{upset}(x, X)$ iff $X = \{y \in V \mid x \leq y\}$ and $t \models \mathrm{downset}(x, X)$ iff $X = \{y \in V \mid y \leq x\}$. We could alternatively write the second of these formulas without the atomic proposition $\mathrm{fin}(X)$ at the expense of additional set quantifications. For later use, we prefer this version. It is not clear to us whether both, set quantifications and atomic propositions $\mathrm{fin}(X)$ can be avoided when expressing $\mathrm{downset}(x, X)$. In the following formulas, we will write $X = \downarrow x$ and $X = \uparrow x$ as a more intuitive abbreviation for the formulas $\mathrm{downset}(x, X)$ and $\mathrm{upset}(x, X)$.

An $\mathrm{MSO}(\mathrm{N}, \lessdot, \mathrm{fin})$-formula is an *$m$-ary modality* if it has m free set variables $X_1, \ldots, X_m$ and one free individual variable x.

Definition 2.2. *An* $\mathrm{MSO}(\mathrm{N}, \lessdot, \mathrm{fin})$-definable temporal logic *is given by*

- *a finite set B of modality names together with a mapping* $\mathrm{arity} : B \to \mathbb{N}$ *giving the arity of each modality name and*
- *a mapping* $[\![-]\!] : B \to \mathrm{MSO}(\mathrm{N}, \lessdot, \mathrm{fin})$ *such that $[\![M]\!]$ is an m-ary modality whenever* $\mathrm{arity}(M) = m$.

Then the syntax of the temporal logic $\mathrm{TL}(B)$ *is defined by the grammar*

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

Let $t = (V, \leq, \lambda)$ be a trace over some finite dependence alphabet (Σ, D) and $\varphi \in \mathrm{TL}(B)$ a formula of $\mathrm{TL}(B)$. The semantics φ^t of φ in t is the set of positions in V where φ holds. The inductive definition is as follows. If $\varphi = a \in \mathrm{N}$, then $\varphi^t = \{x \in V \mid \lambda(x) = a\}$. If $\varphi = M(\varphi_1, \ldots, \varphi_m)$ where $M \in B$ is of arity $m \geq 0$, then

$$\varphi^t = \{p \in V \mid t \models [\![M]\!](\varphi_1^t, \ldots, \varphi_m^t, p)\}.$$

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

For notational convenience and consistency, we consider elements of N as modality names as well and write $[\![a]\!] = (\lambda(x) = a)$ for $a \in \mathrm{N}$.

This definition of an $\mathrm{MSO}(\mathrm{N}, \lessdot, \mathrm{fin})$-definable temporal logic is very much in the style of [7]. It differs in as far as we allow set quantifications and the atomic proposition $\mathrm{fin}(X)$ in our modalities. On the other hand, we do not allow to use the order relation $\leq$ explicitly (but implicitly using set quantification).

Example 2.3. First, the boolean connectives negation and conjunction can be expressed by $[\![\neg]\!](X_1, x) = \neg(x \in X_1)$ and $[\![\wedge]\!](X_1, X_2, x) = (x \in X_1) \wedge (x \in X_2)$.

Existential next $\mathsf{EX}\,\varphi$ is one of the simplest temporal modality. Intuitively, $\mathsf{EX}\,\varphi$ means that there is an immediate successor of the current vertex where φ holds. Formally, we can set $[\![\mathsf{EX}]\!](X_1, x) = \exists y(x \lessdot y \wedge X_1(y))$ which is even a first-order formula.

The unary modality $\mathsf{Eco}\,\varphi$ ("*concurrent*") claims that φ holds for some vertex concurrent to the current vertex x. Thus, its semantics can be defined as

$$[\![\mathsf{Eco}]\!](X_1, x) = \exists X \exists Y \exists z((X = \uparrow x) \wedge (Y = \downarrow x) \wedge z \notin X \cup Y \wedge z \in X_1)\ .$$

Universal strict until $\varphi \,\mathsf{SU}\, \psi$ is a binary modality claiming the existence of a vertex y in the strict future of the current one x such that ψ holds at y and φ holds for all vertices strictly between x and y. Formally, $[\![\mathsf{SU}]\!](X_1, X_2, x)$ is given by

$$\begin{aligned}&\exists X \exists H\ (X = \uparrow x) \wedge H \cap X_2 \neq \emptyset \wedge \forall y \\ &\qquad (y \in H \leftrightarrow y \in X \setminus \{x\} \wedge \forall z \in X(z \lessdot y \rightarrow z \in \{x\} \cup (H \cap X_1))).\end{aligned}$$

The second line expresses that H contains precisely those nodes $y \in X \setminus \{x\}$ such that any vertex strictly between x and y belongs to X_1. Hence, by $H \cap X_2 \neq \emptyset$, there is one such vertex belonging to X_2. The classical non strict version of universal until is $\varphi \,\mathsf{U}\, \psi = \psi \vee (\varphi \wedge (\varphi \,\mathsf{SU}\, \psi))$.

Existential until $\varphi \,\mathsf{EU}\, \psi$ is another binary modality. It claims the existence of some finite path starting in the current node. At any node along this path, φ holds while ψ holds at the final node. Formally, we have

$$\begin{aligned}[\![\mathsf{EU}]\!](X_1, X_2, x) = \exists P, \quad & P \cap X_2 \neq \emptyset \wedge P \subseteq X_1 \cup X_2 \\ & \wedge \forall z \in P,\ (z = x \vee \exists p \in P,\ p \lessdot z).\end{aligned}$$

For more examples, see [8] where most modalities met in the literature on local temporal logics for traces are expressed in terms of $\mathrm{MSO}(\mathrm{N}, \leq)$-modalities. As $\leq$ can be expressed using $\lessdot$, any of those formulas can be transformed into an equivalent one from $\mathrm{MSO}(\mathrm{N}, \lessdot)$.

Uniform satisfiability problem for temporal logics
Let $\mathrm{TL}(B)$ be an $\mathrm{MSO}(\mathrm{N}, \lessdot, \mathrm{fin})$-definable temporal logic.
input: a finite dependence alphabet (Σ, D) and a formula φ of $\mathrm{TL}(B)$
question: Is there a trace $t \in \mathbb{R}(\Sigma, D)$ and a position p in t with $t, p \models \varphi$?

In [8], we considered the non-uniform satisfiability problem for temporal logics definable in $\mathrm{MSO}(\mathrm{N}, \leq)$ where the dependence alphabet (Σ, D) was fixed and not part of the input. By the above discussion, any $\mathrm{MSO}(\mathrm{N}, \lessdot)$-definable temporal logic is $\mathrm{MSO}(\mathrm{N}, \leq)$-definable. Hence [8, Thm. 9] translates into:

Theorem 2.4 ([8]). *The non-uniform satisfiability problem of any* $\mathrm{MSO}(\mathrm{N}, \lessdot)$*-definable temporal logic and any finite dependence alphabet* (Σ, D) *is in* PSPACE.

Analyzing the proof of this result, one obtains the following

Theorem 2.5 (cf. [8]). *For any* $\mathrm{MSO}(\mathrm{N}, \lessdot)$*-definable temporal logic, the uniform satisfiability problem is elementarily decidable.*

In this paper, we present a lower bound and a more precise upper bound for the uniform satisfiability problem. These bounds are expressed in terms of the number of monadic quantifier alternations in the formulas $[\![M]\!]$. Following [6], $\mathrm{M}\Sigma^1_n(\mathrm{N}, \lessdot, \mathrm{fin})$ comprises all $\mathrm{MSO}(\mathrm{N}, \lessdot, \mathrm{fin})$-formulae that are logically equivalent to one of the form $\exists \overrightarrow{X_1} \forall \overrightarrow{X_2} \dots \exists/\forall \overrightarrow{X_n} \varphi$ where φ does not contain any second-order quantification. A formula belongs to $\mathrm{M}\Pi^1_n(\mathrm{N}, \lessdot, \mathrm{fin})$ iff its negation is an element of $\mathrm{M}\Sigma^1_n(\mathrm{N}, \lessdot, \mathrm{fin})$. Finally, $\mathrm{M}\Delta^1_n(\mathrm{N}, \lessdot, \mathrm{fin}) = \mathrm{M}\Sigma^1_n(\mathrm{N}, \lessdot, \mathrm{fin}) \cap \mathrm{M}\Pi^1_n(\mathrm{N}, \lessdot, \mathrm{fin})$. The fragments $\mathrm{M}\Sigma^1_n(\mathrm{N}, \lessdot)$ etc. are defined similarly. Finally, we write $\mathrm{FO}(\mathrm{N}, \lessdot)$ for $\mathrm{M}\Delta^1_0(\mathrm{N}, \lessdot) = \mathrm{M}\Sigma^1_0(\mathrm{N}, \lessdot) = \mathrm{M}\Pi^1_0(\mathrm{N}, \lessdot)$, i.e., for those formulas that can be written without set quantification or $\mathrm{fin}(X)$. In this sense, we speak of $\mathrm{M}\Pi^1_n(\mathrm{N}, \lessdot, \mathrm{fin})$-definable temporal logics whenever all modalities $[\![M]\!]$ belong to $\mathrm{M}\Pi^1_n(\mathrm{N}, \lessdot, \mathrm{fin})$.

Example 2.6. We show that all the modalities EX, Eco, EU, and SU are actually definable in $\mathrm{M}\Delta^1_1(\mathrm{N}, \lessdot, \mathrm{fin})$. Above, we already gave $\mathrm{M}\Sigma^1_1(\mathrm{N}, \lessdot, \mathrm{fin})$-definitions for their semantics, so it remains to present equivalent $\mathrm{M}\Pi^1_1(\mathrm{N}, \lessdot, \mathrm{fin})$-formulas. This is trivial for EX since the $\mathrm{FO}(\mathrm{N}, \lessdot)$-formula $[\![\mathsf{EX}]\!]$ belongs to $\mathrm{M}\Delta^1_1(\mathrm{N}, \lessdot, \mathrm{fin})$. The negation of $[\![\mathsf{Eco}]\!]$ is equivalent to

$$\exists X \exists Y (X = {\uparrow} x \wedge Y = {\downarrow} x \wedge X_1 \subseteq X \cup Y)$$

stating that all nodes z satisfying φ are comparable with x.

To express $\neg[\![\mathsf{SU}]\!]$ by a formula in $\mathrm{M}\Sigma^1_1(\mathrm{N}, \lessdot)$ just replace $H \cap X_2 \neq \emptyset$ by $H \cap X_2 = \emptyset$ in the formula $[\![\mathsf{SU}]\!]$.

Next note that $\neg[\![\mathsf{EU}]\!]$ states that no path in X_1 starts in x and leads to a node in X_2. In other words, the connected component of $({\uparrow} x \cap X_1, \lessdot)$ containing x does not contain any element with upper neighbor in X_2. This can be expressed by the following formula

$$\begin{aligned} \exists X \quad & \forall y \ (y \in X \leftrightarrow y \in X_1 \wedge (y = x \vee \exists z \ (z \in X \wedge z \lessdot y))) \\ & \wedge \forall z \ ((z = x \vee \exists y \ (y \in X \wedge y \lessdot z)) \rightarrow z \notin X_2). \end{aligned}$$

3 nEXPSPACE upper bound for $\mathrm{M}\boldsymbol{\Delta}^{\mathbf{1}}_{\boldsymbol{n-1}}(\mathbf{N},\lessdot,\mathbf{fin})$-logics

The function $\mathtt{tower}:\mathbb{N}\to\mathbb{N}$ is defined inductively by $\mathtt{tower}(0,m)=m$ and $\mathtt{tower}(\ell,m)=2^{\mathtt{tower}(\ell-1,m)}$ for $\ell>0$. It is the aim of this section to prove an upper bound for the uniform satisfiability problem sharper (and more general since it also deals with $\mathrm{MSO}(\mathrm{N},\lessdot,\mathrm{fin})$-modalities) than that given in Theorem 2.5:

Theorem 3.1. *Let* TL *be some* $\mathrm{M}\Delta^1_n(\mathrm{N},\lessdot,\mathrm{fin})$*-definable temporal logic. Then the uniform satisfiability problem for* TL *can be solved in space* $\mathtt{poly}(|\varphi|)\cdot\mathtt{tower}(n+1,\mathtt{poly}(|\Sigma|))$.

The decision procedure we propose refines ideas from [8]. The main ingredient are "modality automata" defined below. Let $w=a_0a_1\dots$ be a word over Σ and $X_i\subseteq\omega$ be sets for $1\leq i\leq m$. Then $(w,\overrightarrow{X})$ denotes the word $b_0b_1\dots$ over $\Sigma\times\{0,1\}^m$ with $b_i=(a_i,x_i^1,x_i^2,\dots,x_i^m)$ and $x_i^j=1$ iff $i\in X_j$.

Definition 3.2. *Let* (Σ,D) *be a finite dependence alphabet and* α *an* m*-ary* $\mathrm{MSO}(\mathrm{N},\lessdot,\mathrm{fin})$*-modality. A Büchi-automaton* $\mathcal{A}$ *over* $\Sigma\times 2^{m+1}$ *is called* modality automaton for (α,Σ,D) *if it accepts precisely those words* $(w,Y_0,Y_1,\dots,Y_m)$ *such that the induced trace* $[w]$ *satisfies* $[w]\models\forall x(\alpha(Y_1,Y_2,\dots,Y_m,x)\leftrightarrow Y_0(x))$.

Before we explain how to use modality automata to solve the uniform satisfiability problem, we fix some more notation: Let φ and ψ be $\mathrm{TL}(B)$-formulas. Then $\mathrm{top}(\varphi)$ denotes the outermost modality name of φ. We write $\varphi\leq\psi$ if φ is a subformula of ψ (this includes the case $\varphi=\psi$). Furthermore $\mathrm{Sub}(\psi)=\{\varphi\in\mathrm{TL}(B)\mid\varphi\leq\psi\}$ is the set of subformulas of ψ. For an alphabet Σ, we will consider words of the form $(w,(Y_\varphi)_{\varphi\in\mathrm{Sub}(\psi)})$ with $w\in\Sigma^\omega$ and $Y_\varphi\subseteq\omega$, i.e., words over the extended alphabet $\Sigma_\psi=\Sigma\times\{0,1\}^{\mathrm{Sub}(\psi)}$. For a subformula $\varphi=M(\varphi_1,\dots,\varphi_m)\leq\psi$ and $\bar{w}=(w,(Y_\xi)_{\xi\leq\psi})\in\Sigma_\psi^\omega$, let $\bar{w}\restriction\varphi=(w,Y_\varphi,Y_{\varphi_1},\dots,Y_{\varphi_m})$.

Now let ξ be some $\mathrm{TL}(B)$-formula and (Σ,D) some finite dependence alphabet. Furthermore, suppose we are given modality automata $\mathcal{A}_M$ for $([\![M]\!],\Sigma,D)$ with set of states Q_M. From these modality automata, we can construct an automaton $\mathcal{A}$ over Σ_ξ with set of states $Q=\prod_{\varphi\leq\xi}Q_{\mathrm{top}(\varphi)}$ that has the following useful property (this construction follows [8] and can alternatively be found in the technical report):

Lemma 3.3. *Let* $\bar{w}=(w,(Y_\varphi)_{\varphi\leq\xi})\in\Sigma_\xi^\omega$*. Then,* $\bar{w}$ *is accepted by* $\mathcal{A}$ *if and only if for each* $\varphi\leq\xi$ *we have* $Y_\varphi=\varphi^{[w]}=\{p\in\omega\mid[w],p\models\varphi\}$.

As an immediate consequence, we obtain

Proposition 3.4. *Let* $w\in\Sigma^\omega$*. Then there exists* $p\in\omega$ *with* $[w],p\models\xi$ *iff there exist* $Y_\varphi\subseteq\omega$ *for* $\varphi\leq\xi$ *with* $Y_\xi\neq\emptyset$ *such that* $(w,(Y_\varphi)_{\varphi\leq\xi})$ *is accepted by* $\mathcal{A}$.

Thus, the satisfiability of ξ is (essentially) equivalent to the emptiness problem for the automaton $\mathcal{A}$. To solve it, we only need to keep in memory three $|\xi|$-tuples of states of our modality automata. Thus, once the modality automata are computed, the satisfiability of ξ can be solved easily. To prove Thm. 3.1, we have to show that a modality automaton for an $\mathrm{M}\Delta^1_n(\mathrm{N},\lessdot,\mathrm{fin})$-definable modality can be constructed in space $\mathtt{tower}(n+1,\mathtt{poly}(|\Sigma|))$ (see Prop. 3.9). Our construction relies on a locality theorem by Schwentick & Bartelmann [12]. In essence, it says that a $\mathrm{FO}(\mathrm{N},\lessdot)$-formula is effectively equivalent to the existence of some finite sets such that any sphere in the structure extended by these "colors" satisfies some first-order property (this is the reason why we consider spheres in traces in the following section). More precisely, this locality theorem holds for connected structures, only.[3] Therefore, from now on, we consider only rooted traces: Suppose there is $\# \in \Sigma$ with $\Sigma \times \{\#\} \subseteq D$. Then $\#\mathbb{M}(\Sigma, D)$ is the set of finite traces over (Σ, D) that have a least node labeled $\#$. Similarly, $\#\mathbb{R}(\Sigma, D)$ comprises all infinite traces over (Σ, D) with such a minimal node. We refer to the elements of $\#\mathbb{R}(\Sigma, D)$ as *rooted traces*. The *uniform rooted satisfiability problem for temporal logics* is the variant of the uniform satisfiability problem where we ask for the existence of some rooted trace. We only prove the above main result for this uniform rooted satisfiability problem, the general case can easily be derived.

3.1 Spheres

The *trace graph* of a trace $t = (V, \leq, \lambda)$ is the structure $G(t) = (V, \leq, \lessdot, (P_a)_{a\in\Sigma})$ given by $P_a = \lambda^{-1}(a)$. The *restriction* of a structure $\mathcal{M} = (W, \leq, \lessdot, (P_a)_{a\in\Sigma})$ to $X \subseteq W$ is the structure

$$\mathcal{M}\restriction X = (X, \leq \cap X^2, \lessdot \cap X^2, (P_a \cap X)_{a\in\Sigma}) \ .$$

If $\mathcal{M} = G(t)$ is a trace graph, $\mathcal{M}\restriction X$ need not be a trace graph itself. In particular, the relation $\lessdot$ in $\mathcal{M}\restriction X$ need not be the covering relation of $\leq$. A *path of length* n in $\mathcal{M}$ is a sequence $x_0, x_1, \ldots, x_n$ with $x_i \in W$ and $(x_i, x_{i+1}) \in (\lessdot \cup \gtrdot)$, i.e., consecutive elements are related by $\lessdot$ in any direction. For $x, y \in W$, the *distance* $d_{\mathcal{M}}(x, y)$ is the minimal length of a path $x_0, \ldots, x_n$ with $x = x_0$ and $y = x_n$. The distance is generalized to $x \in W$ and $U \subseteq W$ by $d_{\mathcal{M}}(x, U) = \min\{d_{\mathcal{M}}(x, y) \mid y \in U\}$. For $r \in \mathbb{N}$ and $U \subseteq W$, let $S_r(\mathcal{M}, U) = \{x \in W \mid d_{\mathcal{M}}(x, U) \leq r\}$ of all elements of W whose distance to U is at most r. Then the *sphere* $\mathrm{Sph}_r(\mathcal{M}, U)$ *around* U denotes the substructure $\mathcal{M}\restriction S_r(\mathcal{M}, U)$.

Let $t = (V, \leq, \lambda)$ be a trace and $U \subseteq V$. Then we write $\mathrm{Sph}_r(t, U)$ for $\mathrm{Sph}_r(G(t), U)$. Furthermore, for $a \in \mathrm{alph}(t)$, let $\mathrm{last}_a(t) = \max(\lambda^{-1}(a))$ be the $\leq$-maximal a-labeled node occurring in t. Let $\mathrm{last}(t) = \{\mathrm{last}_a(t) \mid a \in \mathrm{alph}(t)\}$ and for $r \in \mathbb{N}$, let $\mathrm{top}_r(t)$ be the structure $\mathrm{Sph}_r(t, \mathrm{last}(t))$, i.e., the restriction of $G(t)$ to those nodes from t whose distance to some maximal a-labeled node is at most r.

[3] Hanf and Gaifman proved similar locality theorems (cf. [6]). We could not use Hanf's theorem since there is no uniform bound for the degree of trace graphs independent from the dependence alphabet. Using Gaifman's theorem results in slightly more involved automata constructions (because of the disjointness condition) without improving the result.

Example 3.5. Let $\Sigma = \{a, b, c, d\}$ with $I = \{(b, d), (d, b), (a, c), (c, a)\}$ and consider the trace $s = [aabbcccbbbb]$. In Fig. 1, the trace graph of $s \cdot d$ is depicted in the first line. There, solid edges denote the covering relation $\lessdot$. Furthermore, black nodes are those in $\mathrm{last}(sd)$. In the second picture, the structure $\mathrm{top}_1(sd)$ is depicted. There, solid arrows have the same meaning as in the first picture, but the partial order relation $\leq$ is the reflexive and transitive closure of all arrows (including the dashed ones). If, in this second picture, we erase the d-labeled node, we obtain $\mathrm{top}_1(s)$. Note the similarity of these pictures with those of Fig. 2 with $t = [ccbbaaabbbb]$: In particular, the covering relation restricted to $\mathrm{top}_1(s)$ and $\mathrm{top}_1(t)$ are equal, but they differ in $\mathrm{top}_1(sd)$ and $\mathrm{top}_1(td)$. Thus, although we are only interested in the relation $\lessdot$, in order to update this information, we also have to keep the order in the top sphere. The following lemma shows that this information is sufficient to compute $\mathrm{top}_r(sd)$ from $\mathrm{top}_r(s)$.

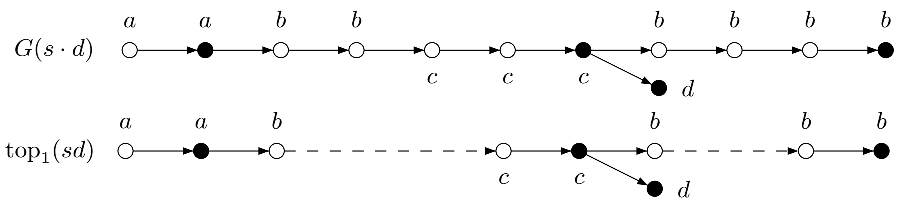

Fig. 1. Update of $\mathrm{top}_1(s)$

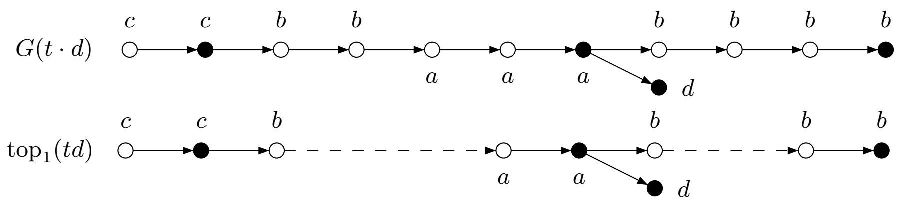

Fig. 2. Update of $\mathrm{top}_1(t)$

Lemma 3.6. *Let s be a trace, $a \in \Sigma$, and $r \in \mathbb{N}$. Then $\mathrm{top}_r(sa)$ is determined by $\mathrm{top}_r(s)$ and the letter a.*

Let $w \in \Sigma^\infty$ and $t = [w] = (V, \leq, \lambda) \in \mathbb{R}(\Sigma, D)$. Fix also some $r \in \mathbb{N}$. A modality automaton will have to check properties of spheres of the form $\mathrm{Sph}_r(t, x)$. For each $x \in V$, we can find a finite prefix u of w such that $\mathrm{Sph}_r(t, x)$ is contained in $\mathrm{top}_{2r}([u])$. From Lemma 3.6, the structures $\mathrm{top}_{2r}([u])$ can be computed by an automaton. But we also need to determine when a vertex x in $\mathrm{top}_{2r}([u])$ is such that $\mathrm{Sph}_r(t, x)$ is contained in $\mathrm{top}_{2r}([u])$. This is the purpose of the following definition and lemma.

Definition 3.7. *Let $s = (V, \leq, \lambda) \in \mathbb{M}(\Sigma, D)$ be a finite trace. Let $B \subseteq \Sigma$ and $r \in \mathbb{N}$. A vertex $x \in V$ is* r-critical *for (s, B) if $d_{G(s)}(x, \mathrm{last}(s)) \leq r$, and for all*

$(a,b) \in D \cap (\mathrm{alph}(s) \times B)$, *if* $d_{G(s)}(x, \mathrm{last}_a(s)) < r$ *then* $\mathrm{last}_a(s) < \mathrm{last}_c(s)$ *for some* $c \in \mathrm{alph}(s)$ *with* $(c,b) \in D$.

Note that we can determine whether x is r-critical for (s,B) just knowing $\mathrm{top}_r(s)$ and B.

Lemma 3.8.

1. *Let* $s = (V, \leq, \lambda) \in \mathbb{M}(\Sigma, D)$, $B \subseteq \Sigma$ *and* $r \in \mathbb{N}$. *If a vertex* $x \in V$ *is* r*-critical for* (s, B) *then for all* $t \in \mathbb{R}(\Sigma, D)$ *with* $\mathrm{alph}(t) \subseteq B$, *we have* $\mathrm{Sph}_r(st, x) = \mathrm{Sph}_r(\mathrm{top}_{2r}(s), x)$.
2. *Let* $w \in \Sigma^\infty$, $[w] = (V, \leq, \lambda) \in \mathbb{R}(\Sigma, D)$, $x \in V$ *and* $r \in \mathbb{N}$. *There is a factorisation* $w = uv$ *with* u *finite such that* x *is* r*-critical for* $([u], \mathrm{alph}(v))$.

3.2 Construction of modality automata

For a k-ary $\mathrm{M}\Delta^1_n(\mathrm{N}, \lessdot, \mathrm{fin})$-modality α, the formula $\forall x(\alpha(Y_1, \ldots, Y_k, x) \leftrightarrow Y_0(x))$ belongs to $\mathrm{M}\Pi^1_n(\mathrm{N}, \lessdot, \mathrm{fin})$. Its kernel can be written as a Boolean combination of $\mathrm{FO}(\mathrm{N}, \lessdot)$-formulas $\varphi(X_1, \ldots, X_m)$ and formulas of the form $\mathrm{fin}(X)$. By [12, Theorem 3.2(2)], there are $\ell, r \in \mathbb{N}$ and a formula $\psi \in \mathrm{FO}(\mathrm{N}, \lessdot)$ with $h = m + \ell$ many free set variables $X_1, \ldots, X_{m+\ell}$ and one free individual variable y such that

- ψ is r*-local around* y (i.e., quantifications in ψ are restricted to nodes z with $d(y,z) \leq r$)
- for any finite dependence alphabet (Σ, D) and for any rooted trace $t = (V, \leq, \lambda) \in \#\mathbb{R}(\Sigma, D)$ together with m sets $X_1, \ldots, X_m$ contained in V, we have $(t, X_1, \ldots, X_m) \models \varphi$ iff $(t, X_1, \ldots, X_m) \models \exists X_{m+1} \ldots \exists X_{m+\ell} (\forall y\, \psi \wedge \bigwedge_{1 \leq i \leq \ell} \mathrm{fin}(X_{m+i}))$.

Let (Σ, D) be some finite dependence alphabet. We define a Büchi-automaton over the alphabet $\Sigma \times 2^h$ as follows. A state of this automaton is a tuple $q = (\mathrm{top}_{2r}(s), (X_i)_{1 \leq i \leq h}, B, C)$ where

- $s = (V, \leq, \lambda) \in \#\mathbb{M}(\Sigma, D)$ is some finite rooted trace and X_i is contained in $\mathrm{top}_{2r}(s)$ for $1 \leq i \leq h$,
- $B, C \subseteq \Sigma$, (the intuition is that B is used to guess the alphabet of the word that remains to be read and C is used to check the correctness of this guess)
- for any y in $\mathrm{top}_r(s)$ which is r-critical for (s, B), if we let Y_i be the intersection of X_i and $\mathrm{Sph}_r(\mathrm{top}_{2r}(s), y)$ for $1 \leq i \leq h$, then we have $\mathrm{Sph}_r(\mathrm{top}_{2r}(s), y) \models \psi(Y_1, \ldots, Y_h, y)$.

Based on this set of states Q and using Lemmas 3.6 and 3.8, we can then define a *sphere automaton* $\mathcal{A}(\Sigma, D, \psi)$ in space $2^{h|\Sigma|^{O(r)}}$ that checks, reading a word $\overline{w} = (w, X_1, \ldots, X_h)$ with $w \in \#\Sigma^\omega$ whether $([w], X_1, \ldots, X_h) \models \forall y \psi$. In this automaton, if we have a transition labeled $(a, x_1, \ldots, x_h)$ between states $q = (\mathrm{top}_{2r}(s), (X_i)_{1 \leq i \leq h}, B, C)$ and $q' = (\mathcal{M}', (X'_i)_{1 \leq i \leq h}, B', C')$ then we have $\mathcal{M}' = \mathrm{top}_{2r}(sa)$, $B = B' \cup \{a\}$, $C' = C \setminus \{a\}$ if $C \neq \emptyset$ and $C' = B'$ otherwise, and for each $1 \leq i \leq \ell$, the set X'_i is obtained from X_i by removing the vertices in $\mathrm{top}_{2r}(s)$ that are no longer in $\mathrm{top}_{2r}(sa)$ and by adding a new vertex if $x_1 = 1$.

Note that the finiteness of a set X can be checked by a fixed automaton. Applying the usual Boolean operations and projections to these Büchi-automata, we obtain the following proposition which completes the proof of Theorem 3.1.

Proposition 3.9. *Let α be an $\mathrm{M}\Delta^1_n(\mathrm{N}, \lessdot)$-modality. Then the following problem can be solved in space* $\mathtt{tower}(n+1, \mathtt{poly}(|\Sigma|))$
input: *a finite dependence alphabet (Σ, D)*
output: *a modality automaton for (α, Σ, D).*

4 nEXPSPACE lower bound for $\mathrm{M}\boldsymbol{\Pi}^1_{n+1}(\mathrm{N}, \lessdot)$-logics

This section is devoted to the proof of

Theorem 4.1. *Let $n \in \mathbb{N}$. There is an $\mathrm{M}\Pi^1_{n+1}(\mathrm{N}, \lessdot)$-definable temporal logic TL_{n+1} such that its uniform satisfiability problem is* **nEXPSPACE***-hard.*

Towards this aim, we will restrict ourselves to finite traces, the general result can easily be derived.

Idea of proof and notation: Let M be a deterministic Turing machine working in space $\mathtt{tower}(n, m)$ where m is the length of the input word. A configuration of M is described by a word $\triangleright\alpha q\beta\triangleleft$ (of length $\mathtt{tower}(n, m)$) over some alphabet Γ where q is the current state, $\alpha\beta$ is the tape contents and the head of M is on the first letter of β. We write $w \vdash w'$ if there is a transition of M from the configuration w to the configuration w'. To encode the computation $(w_0, w_1, \ldots, w_k)$ of M, we consider the word $c_{n+1}w_0d_{n+1}\, c_{n+1}w_1d_{n+1}\, \ldots c_{n+1}w_kd_{n+1}$ where c_{n+1} and d_{n+1} are new letters that act as delimiter. Now, to relate consecutive configurations, we will add counters that describe the index in the configuration (i.e., any configuration gets replaced by an alternating sequence of letters and counters). These counters will use additional letters and the number of additional letters used determines an upper bound of the value of these counters. Our main task in this section will be to encode counters that can count up to $\mathtt{tower}(n, m)$ using only linearly many new letters (in m). In a first step, we will encode these counters using new letters from an infinite set A. This will be achieved using in addition to the covering relation $\lessdot$ a relation $\prec$ that is an appropriate restriction of $\sqsubset$ from the previous section. Thus, in Section 4.1, we will describe encodings of successful computations of M as words using the relations $\lessdot$ and $\prec$. In the following Section 4.2, we will consider traces over some larger alphabet. This larger alphabet will allow us to replace the relation $\prec$ just using the covering relation $\lessdot$. Thus, we will be able to encode successful computations in traces. The remaining procedure (to be found in Section 4.3) is standard: from an input word v of length m, we will define a formula φ of the temporal logic TL (that we are going to construct from the Turing machine M) and an alphabet (Σ_m, D) of size $O(m)$ such that φ is satisfiable in $\mathbb{M}(\Sigma_m, D)$ iff M accepts the word v.

4.1 Encoding by words

Notation We fix some pairwise disjoint alphabets Γ (the "alphabet" of the Turing machine M), $B_i = \{0_i, 1_i\}$ and $C_i = \{c_i, d_i\}$ for $i > 0$ and we let $\Delta_i = \Gamma \cup \bigcup_{j \geq i}^{n} B_j \cup \bigcup_{j \geq i}^{n+1} C_j$. Furthermore, let A be an infinite set and $\Sigma = A \uplus \Delta_1$. For a set $E \subseteq \Sigma$ of letters, let $\Pi_E : \Sigma^* \to E^*$ be the projection to E (we will use this in particular for $E = B_i, A, \Gamma$). For Π_{B_i}, we write simply Π_i.

In this section, we consider formulas of the logic $\mathrm{MSO}(\lessdot, \prec)$ that speak about words over the alphabet Σ. Atomic formulas are of the form $x \in X$, $\lambda(x) = a$ for $a \in \Delta_1$, $x \lessdot y$, and $x \prec y$. We define the semantics of $\prec$ by $w \models x \prec y$ iff

$$x < y \wedge (\lambda(x), \lambda(y) \in A \to \lambda(x) = \lambda(y)) \wedge \forall z (x < z < y \to \lambda(z) \notin \{\lambda(x), \lambda(y)\}).$$

We will freely use formulas like $\lambda(x) \in E$ for $E \subseteq \Delta_1$ meaning $\bigvee_{e \in E} \lambda(x) = e$. Note that formulas $\lambda(x) = a$ for $a \in A$ are not allowed, but we use $\lambda(x) \in A$ for $\neg(\lambda(x) \in \Delta_1)$.

Level 1 counters: Let $K_1 = c_1(AB_1)^+ d_1$. The intuition is that a word $v = c_1 a_1 b_1 \cdots a_m b_m d_1$ is a counter whose value is $\Pi_1(v) \in B_1^m$ representing a number between 0 and $2^m - 1$.

Let L_1 be the set of words $v = u_0 v_1 u_1 \cdots v_k u_k \in \Delta_2^*(K_1 \Delta_2^*)^+$ with $v_i \in K_1$ and $u_i \in \Delta_2^*$ such that $\Pi_A(v_i)$ uses any letter at most once and $\Pi_A(v_i) = \Pi_A(v_j)$ for all $1 \leq i, j \leq k$. In other words, L_1 is the union of all languages $\Delta_2^*(c_1 a_1 B_1 \cdots a_m B_1 d_1 \Delta_2^*)^+$ where $a_1, \ldots, a_m$ is a sequence of pairwise distinct letters from A. Here the intuition is that a word $w \in L_1$ is a sequence of Δ_2 words separated by level 1 counters all using the same number m of bits for some $m > 0$.

Lemma 4.2. *The language L_1 can be defined in $\mathrm{FO}(\lessdot, \prec)$, i.e., there is a sentence $\varphi_1 \in \mathrm{FO}(\lessdot, \prec)$ such that $L_1 = \{w \in \Sigma^* \mid w \models \varphi_1\}$.*

Proof. First, one writes a formula expressing that a word belongs to $\Delta_2^*(K_1 \Delta_2^*)^+$. Since we can express that the letter at position x belongs to A (by saying that it does not belong to the finite set Δ_1), this is clearly possible. Next, one has to express that no factor from K_1 contains any letter from A more than once. If A was finite, this could be achieved by listing all these requirements. The problem caused by the infinity of A can be solved by saying that any position x with $\lambda(x) \in A$ is $\prec$-related to some d_1-position: $\forall x \exists y (\lambda(x) \in A \to \lambda(y) = d_1 \wedge x \prec y)$. Finally, factors from K_1 shall contain the same sequence of letters of A. By transitivity, it suffices to express this for consecutive such factors that start and end, resp., in positions x, z and x', z' with $\lambda(x) = \lambda(x') = c_1$, $\lambda(z) = \lambda(z') = d_1$, and $x \prec z \prec x' \prec z'$. Then, the formula requires that the relation $\prec$, restricted to the A-positions between x and z is an order-isomorphism (more precisely: $\prec$-isomorphism) onto the A-positions between x' and z'. ◻

Level ℓ counters ($1 < \ell \leq n$): Let K_ℓ be the set of words $v \in L_{\ell-1} \cap c_\ell(K_{\ell-1} B_\ell)^+ d_\ell$ such that if we write $v = c_\ell v_0 b_0 \cdots v_k b_k d_\ell$ with $v_i \in K_{\ell-1}$ and $b_i \in B_\ell$ then we have $\Pi_{\ell-1}(v_0) \in 0_{\ell-1}^+$, $\Pi_{\ell-1}(v_k) \in 1_{\ell-1}^+$, and $\Pi_{\ell-1}(v_{i+1}) =$

$\mathrm{succ}(\Pi_{\ell-1}(v_i))$ for all $0 \le i < k$, where succ denotes the successor for the lexicographic order. As above, the intuition is that a word $v \in K_\ell$ is a counter whose value is $\Pi_\ell(v) \in B_\ell^+$ representing an integer between 0 and $\mathtt{tower}(\ell, m) - 1$ for some $m > 0$. Note that each bit b_i in a level ℓ counter is preceded by a level $\ell - 1$ counter v_i whose value is the index of this bit. Note also that words from K_ℓ use letters from $A \cup \bigcup_{1 \le i \le \ell}(B_i \cup C_i)$, only.

Now, let $L_\ell = L_{\ell-1} \cap \Delta_{\ell+1}^*(K_\ell \Delta_{\ell+1}^*)^+$. The intuition is that a word in L_ℓ represents a sequence of level ℓ counters all using the same number $\mathtt{tower}(\ell - 1, m)$ of bits for some $m > 0$.

Computation of the Turing Machine M: Recall that M is a deterministic Turing machine working in space $\mathtt{tower}(n, m)$ with m the length of the input.

Let K be the set of words $v \in L_n \cap c_{n+1}(K_n \Gamma)^+ d_{n+1}$ such that if we write $v = c_{n+1} v_0 \gamma_0 \cdots v_k \gamma_k d_{n+1}$ with $v_i \in K_n$ and $\gamma_i \in \Gamma$ then $\Pi_n(v_0) \in 0_n^+$, $\Pi_n(v_k) \in 1_n^+$, and $\Pi_n(v_{i+1}) = \mathrm{succ}(\Pi_n(v_i))$ for all $0 \le i < k$. As above, the intuition is that a word $v \in K$ describes a configuration of M whose value is $\Pi_\Gamma(v) \in \Gamma^+$. Note that each letter γ_i in a configuration is preceded by a level n counter v_i whose value is the index of this letter.

Now, let L be the set of words $w \in L_n \cap K^+$ such that if we write $w = w_0 w_1 \cdots w_k$ with $w_i \in K$ then we have $\Pi_\Gamma(w_i) \vdash \Pi_\Gamma(w_{i+1})$ for all $0 \le i < k$. In other words, the language L is the set of encodings of computations of M.

Lemma 4.3. *The language L can be defined in $\mathrm{M}\Pi^1_{n+1}(\lessdot, \prec)$, i.e., there is a sentence $\psi \in \mathrm{M}\Pi^1_{n+1}(\lessdot, \prec)$ such that $L = \{w \in \Sigma^* \mid w \models \psi\}$.*

4.2 From words to traces

We consider a disjoint copy $\bar{A} = \{\overline{a} \mid a \in A\}$ of A and we let $\Sigma' = \Sigma \cup \{\dagger\} \cup \bar{A} \cup (\Sigma \times \Delta_1)$. The dependence relation D on Σ' is the least reflexive and symmetric relation such that elements of $\Sigma \cup \{\dagger\}$ are mutually dependent, $\bar{a}$ depends on a only (for $a \in A$), and (σ, δ) depends on σ and δ only. For simplicity, we write $\mathbb{M}$ for the trace monoid $\mathbb{M}(\Sigma', D)$.

Formulas in this section will mention the actions from $\Delta_1 \cup \{\dagger\} \cup (\Delta_1 \times \Delta_1)$ only. In this section, we will define a set $L' \subseteq \mathbb{M}$ definable in $\mathrm{MSO}(\mathrm{N}, \lessdot)$ such that $\Pi_\Sigma(L') = L$, the language from Lemma 4.3.

For $\sigma \in \Sigma$, we let $T(\sigma)$ be the unique trace having exactly one occurrence of each letter in $\{\sigma\} \times \Delta_1$. Define a homomorphism $\eta : \Sigma^* \to \mathbb{M}(\Sigma', D)$ by

$$\eta(\sigma) = \begin{cases} \bar{a}\, T(a)\, a \dagger \bar{a}\, T(a) & \text{if } \sigma = a \in A \\ \sigma \dagger T(\sigma) & \text{otherwise.} \end{cases}$$

For $\Delta_1 = \{\tau_1, \ldots, \tau_k\}$, the traces $\eta(a)$ and $\eta(\sigma)$ with $a \in A$ and $\sigma \in \Delta_1$ are depicted in Fig. 3. Note that for any $w \in \Sigma^*$, we have $\Pi_{\Sigma \cup \{\dagger\}}(\eta(w)) \in (\Sigma\dagger)^*$. The language L' that we will define here is precisely $\dagger\eta(L)$.

Lemma 4.4. *There is a formula $\varphi \in \mathrm{FO}(\mathrm{N}, \lessdot)$ such that a trace $t \in \mathbb{M}$ satisfies φ iff $t \in \dagger\eta(\Sigma^*)$.*

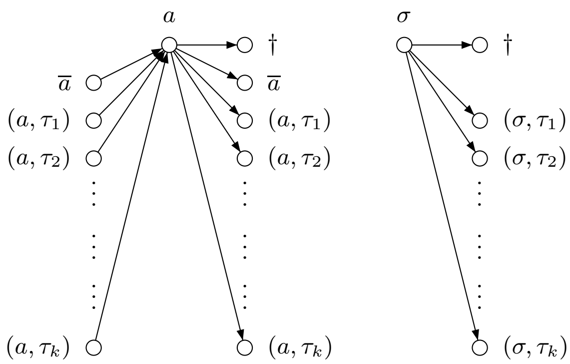

Fig. 3. The traces $\eta(a)$ and $\eta(\sigma)$

Proof (Sketch). We construct $\varphi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3$. The formula φ_1 ensures that $\Pi_{\Sigma\cup\{\dagger\}}(t) \in \dagger(\Sigma\dagger)^*$. The formula φ_2 ensures that any Σ-labeled node is the center of some factor $\eta(\sigma)$. This is easy if $\sigma \in \Delta_1$ since $\{\dagger\} \cup (\Delta_1 \times \Delta_1)$ is finite. For $\sigma \in A$, it turns out to be sufficient to require the existence of at least $|\Delta_1|+2$ many upper and lower neighbors. The formula φ_3 expresses that the whole trace is the disjoint union of these factors. $\square$

For any word $w \in \Sigma^*$ we have $w = \Pi_\Sigma(t)$ where $t = \dagger\eta(w)$. Thus, the word w can be seen as a chain in the trace $t = \dagger\eta(w)$. The predicate $(\lambda(x) \in \Sigma)$ and the relations $\lessdot$ and $\prec$ of w can be expressed in t by $\mathrm{FO}(\mathrm{N}, \lessdot)$-formulas as follows:

$$\begin{aligned}
(\lambda(x) \in \Sigma) &= \exists z, (x \lessdot z) \wedge (\lambda(z) = \dagger),\\
\mathrm{cover}(x,y) &= \exists z, (x \lessdot z \lessdot y) \wedge (\lambda(z) = \dagger),\\
\mathrm{nx}(x,y) &= \exists z_1 \exists z_2, (x \lessdot z_1 \lessdot y) \vee (x \lessdot z_1 \lessdot z_2 \lessdot y).
\end{aligned}$$

More precisely, for $w \in \Sigma^*$, $t = \dagger\eta(w)$ and x, y in t with $\lambda(x), \lambda(y) \in \Sigma$, we have $x \lessdot y$ in w iff $t \models \mathrm{cover}(x,y)$, and $x \prec y$ in w iff $t \models \mathrm{nx}(x,y)$. This allows immediately to derive the following consequence since L is definable in $\mathrm{M}\Pi^1_{n+1}(\lessdot, \prec)$:

Proposition 4.5. *The language $\dagger\eta(L)$ is $\mathrm{M}\Pi^1_{n+1}(\mathrm{N}, \lessdot)$-definable, i.e., there is a sentence $\overline{\psi} \in \mathrm{M}\Pi^1_{n+1}(\mathrm{N}, \lessdot)$ such that $\dagger\eta(L) = \{t \in \mathbb{M} \mid t \models \overline{\psi}\}$.*

4.3 The lower bound

Proof (of Theorem 4.1). Recall that the deterministic Turing machine M works (with an input of length m) in space $\mathtt{tower}(n,m)$.

Consider the $\mathrm{M}\Pi^1_{n+1}$-definable temporal logic TL_{n+1} based on the modality SU, the usual boolean connectives, and the constant COMPUTATION with $[\![\mathrm{COMPUTATION}]\!] = \overline{\psi}$, the formula from Proposition 4.5 defining $\dagger\eta(L)$.

We denote by q_0 and q_1 the initial state and the accepting state of M respectively. We also denote by $\Box$ the blank letter of the tape. Let $v = v_1 \cdots v_m$ be an input word of the Turing machine M and consider the formula INIT_v

$$\neg\Gamma \,\mathsf{SU}\, (\triangleright \wedge \neg\Gamma \,\mathsf{SU}\, (q_0 \wedge \neg\Gamma \,\mathsf{SU}\, (v_1 \wedge \cdots \neg\Gamma \,\mathsf{SU}\, (v_m \wedge (\neg\Gamma \vee \Box) \,\mathsf{SU}\, \triangleleft) \cdots)))$$

which intuitively expresses the fact that the first configuration is actually the initial configuration of M on the input word v. Consider also the alphabets $\Sigma_m = A_m \cup \Delta_1 \subseteq \Sigma$ and $\Sigma'_m = \Sigma_m \cup \overline{A_m} \cup (\Sigma_m \times \Delta_1) \cup \{\dagger\}$ where $|A_m| = m$, $\overline{A_m} = \{\overline{a} \mid a \in A_m\}$ and the dependence relation D defined as above. We claim that v is accepted by M if and only if there is a trace in $\mathbb{M}(\Sigma_m, D)$ satisfying the formula $g_v = \mathrm{COMPUTATION} \wedge \mathrm{INIT}_v \wedge \top \,\mathsf{SU}\, q_1$. Therefore, the uniform satisfiability problem for TL_{n+1} is **nEXPSPACE**-hard. $\Box$

Remark 4.6. Note that all modalities of the logic TL_{n+1} are of arity at most two. Furthermore, the only binary temporal modality is SU. In our hardness proof, it is only used in the context $\neg\Gamma \,\mathsf{SU} -$, $(\neg\Gamma \vee \Box) \,\mathsf{SU} -$ and $\top \,\mathsf{SU} -$. Thus, we could have replaced the binary modality SU by these three unary filter modalities in the style of [9]. Furthermore, the temporal logic could be deprived of constant formulas a for $a \notin \Gamma$ since they are not used in the hardness proof.

5 Process-based $\mathrm{M}\Delta^1_1(\mathrm{N}, \lessdot, \mathrm{fin})$-definable logics

In Sect. 2, we showed that the modalities EX, Eco, EU, and SU can be dealt with in our framework. The technical report that this paper is based on shows that all modalities considered in the context of action based local temporal logics fall into our framework. Hence our upper bound shows that their uniform satisfiability problem can be solved in space $\mathtt{poly}(|\varphi|) \cdot \mathtt{tower}(2, \mathtt{poly}(|\Sigma|))$.

It is the aim of this final section to indicate that also Thiagarajan's logic TrPTL [13] can be dealt with in the setting of $\mathrm{M}\Delta^1_1(\mathrm{N}, \lessdot, \mathrm{fin})$-definable local temporal logics. The underlying idea of TrPTL 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 and finite set of processes $p(a) \subseteq \mathbb{N}$ 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 $\mathcal{P}_i$, $\mathcal{O}_i$ and $\mathcal{U}_i$ ($i \in \mathbb{N}$) of arity 0, 1 and 2 respectively. Intuitively, $\mathcal{P}_i$ 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, $\varphi\mathcal{U}_i\psi$ 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.

We only explain how to handle $\mathcal{P}_i$, the remaining modalities are discussed in the technical report. The basic idea is that the index i is dealt with as an additional argument, i.e., $\mathcal{P}$ is considered to be of arity 1. The semantics of $\mathcal{P}$

is then given by the first-order formula $[\![\mathcal{P}]\!](X_1, x) = (x \in X_1)$. Then, given a concrete and finite dependence alphabet (Σ, D) together with a concrete function $p : \Sigma \to \mathbb{N}$, Thiagarajan's formula $\mathcal{P}_i$ is given by $\mathcal{P}(\bigvee_{a \in p^{-1}(i)} a)$. Thus, a temporal logic admitting modalities $\vee$ and $\mathcal{P}$ is able to simulate the modality $\mathcal{P}_i$. Similarly, we can deal with the other process based modalities.

References

1. B. Adsul and M. Sohoni. Complete and tractable local linear time temporal logics over traces. In *Proc. of ICALP'02*, number 2380 in LNCS, pages 926–937. Springer Verlag, 2002.
2. R. Alur, D. Peled, and W. Penczek. Model-checking of causality properties. In *Proc. of LICS'95*, pages 90–100. IEEE Computer Society Press, 1995.
3. V. Diekert and P. Gastin. Local temporal logic is expressively complete for cograph dependence alphabets. *Information and Computation*, 195:30–52, 2004.
4. V. Diekert and P. Gastin. Pure future local temporal logics are expressively complete for Mazurkiewicz traces. In *Proc. of LATIN'04*, number 2976 in LNCS, pages 232–241. Springer Verlag, 2004.
5. V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, Singapore, 1995.
6. H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer Verlag, 1991.
7. D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic*. Oxford University Press, 1994.
8. P. Gastin and D. Kuske. Satisfiability and model checking for MSO-definable temporal logics are in PSPACE. In *Proc. of CONCUR'03*, number 2761 in LNCS, pages 222–236. Springer Verlag, 2003.
9. 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.
10. P. Gastin, M. Mukund, and K. Narayan Kumar. Local LTL with past constants is expressively complete for Mazurkiewicz traces. In *Proc. of MFCS'03*, number 2747 in LNCS, pages 429–438. Springer Verlag, 2003.
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. Th. Schwentick and K. Bartelmann. Local normal forms for first-order logic with applications to games and automata. *Discrete Mathematics and Computer Science*, 3:109–124, 1999.
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. Springer Verlag, 1995.