Information and Computation 208 (2010) 797–816

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/ic

Uniform satisfiability problem for local temporal logics over Mazurkiewicz traces ☆

Paul Gastin [a], Dietrich Kuske [b,*,1]

[a] LSV, ENS Cachan & INRIA & CNRS, 61, Av. du Président Wilson, F-94235 Cachan Cedex, France
[b] CNRS, LaBRI, Université Bordeaux I, 351, cours de la Libération, F-33405 Talence, France

ARTICLE INFO

Article history:
Received 08 October 2007
Revised 14 April 2009
Available online 11 February 2010

ABSTRACT

We continue our study of the complexity of MSO-definable local temporal logics over concurrent systems that can be described by Mazurkiewicz traces. In previous papers, we showed that the satisfiability problem for any such logic is in **PSPACE** (provided the dependence alphabet is fixed, Gastin and Kuske (2003) [10]) and remains in **PSPACE** for all classical local temporal logics even if the dependence alphabet is part of the input, Gastin and Kuske (2007) [8]. In this paper, we consider the uniform satisfiability problem for arbitrary MSO-definable local temporal logics. For this problem, we prove multi-exponential lower and upper bounds that depend on the number of alternations of set quantifiers present in the chosen MSO-modalities.

1. Introduction

Executions of distributed systems can be modeled as Mazurkiewicz traces [4] 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 15 years, a lot of papers have been devoted to the study of temporal logics over partial orders and in particular over Mazurkiewicz traces (cf. [1–3,5,8–11,19,22,23]). 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 (as opposed to global temporal logics [24]). The common feature of these logics is that formulas are evaluated at single events corresponding to local views of processes. In [10], we proposed a unified treatment of all those local temporal logics that can be presented 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 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. This proof constructs a finite automaton from a formula of the temporal logic such that a word is accepted iff its associated trace satisfies the formula. This construction is similar to the one considered in [14]; it does not rely on alternating automata (a technique that goes back to [20]), but on *modality automata*. The idea can be best explained in terms of transducers: given a formula, we aim at a transducer that marks all positions in a word where the formula holds. Given two such transducers for the formulas φ and ψ and given a modality $M(\varphi, \psi)$ of the temporal logic (e.g., *until* or any

☆ Work partly supported by projects DAAD-PROCOPE, ARCUS Île de France-Inde and ANR-06-SETIN-003 DOTS.
* Corresponding author.
Email addresses: Paul.Gastin@lsv.ens-cachan.fr (P. Gastin), kuske@labri.fr (D. Kuske).
[1] This work was done when the second author was affiliated with Universität Leipzig, Germany.

0890-5401/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2009.12.003

other MSO-definable modality), a modality automaton is a transducer that takes as input the outputs of the transducers for φ and ψ (i.e., it knows for each position in the word which of the two formulas hold) and produces the marking for the combined formula $M(\varphi, \psi)$. These modality automata depend on the architecture of the system, i.e., on the dependence alphabet (Σ, D). But if this dependence alphabet is fixed, their construction does not contribute to the complexity of the satisfiability problem.

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 $\varphi \in \mathrm{TL}$ can be satisfied in a given architecture (Σ, D) (described as a trace alphabet). Differently from the non-uniform case, now the modality automata cannot be computed in a preprocessing step, but their computation contributes to the complexity. In [8], we presented a sufficient condition on the modalities (*polynomial variance*) that allowed an efficient construction of modality automata and therefore an efficient solution of the uniform satisfiability problem. Since, as we also showed, all local temporal logics considered in the literature satisfy our sufficient condition, we obtained that the uniform satisfiability problem for any of them is in **PSPACE** (due to the compositionality of our method, this applies even to the logic that features all modalities considered in the literature).

In this paper, we study the uniform satisfiability problem for arbitrary MSO-definable local temporal logics. Recall that the semantics of the modality names of TL are given by MSO formulas. We prove lower and upper bounds for the complexity of the associated uniform satisfiability problem that depend on the number of alternations of set quantifiers in these modalities. To state our results more precisely, recall that $\mathrm{M}\Pi^1_n$ is the set of MSO-formulas that can be written as $\forall\overrightarrow{X_1}\,\exists\overrightarrow{X_2}\cdots\exists/\forall\overrightarrow{X_n}\,\varphi$ where φ does not contain any set quantifiers, and $\mathrm{BoolM}\Sigma^1_n$ is the set of Boolean combinations of formulas from $\mathrm{M}\Pi^1_n$. If the semantics of every modality name in the local temporal logic TL belongs to $\mathrm{BoolM}\Sigma^1_n$, then the uniform satisfiability problem can be solved in n-fold exponential space (Theorem 3.1 and Remark 2.6). This result is optimal since, for every $n > 0$ we present a local temporal logic TL whose modalities belong to $\mathrm{M}\Pi^1_n$ and whose uniform satisfiability problem is hard (and therefore complete) for n-fold exponential space (Theorem 4.1 and Remark 2.6).

Again, the decision procedure for the upper bound is based on modality automata. Schwentick and Bartelmann [21] give a normal form for the first-order kernel of the MSO-formulas that describe the semantics of modalities. This normal form allows to compute these automata more efficiently than expected (cf. discussion before Proposition 3.4). The lower bound is shown by a reduction of the word problem from an arbitrary Turing machine working in n-fold exponential space. The reduction is based on an adaptation of Matz' method [16,17] of n-fold exponential counting by n monadic quantifier alternations.

An extended abstract presenting weaker results appeared as [12].

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 *finite* set of action names and the dependence relation $D \subseteq \Sigma^2$ is symmetric and reflexive. The independence relation is $I = \Sigma^2 \setminus D$. For $A \subseteq \Sigma$, we let $D(A) = \{b \in \Sigma \mid (a, b) \in D \text{ for some } a \in A\}$ be the set of letters that depend on some letter in A, and we let $I(A) = \Sigma \setminus D(A)$ be the set of letters independent from all letters in A. A *trace over* (Σ, D) is a labeled at most countably infinite partial order $t = (V, \leqslant, \lambda)$ such that $(V, \leqslant)$ is a partial order and $\lambda : V \to \Sigma$ is the labeling function satisfying for all $x, y \in V$

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

where $\lessdot = {<} \setminus {<}^2$ is the immediate successor relation. The alphabet of the trace t is $\mathrm{alph}(t) = \lambda(V)$ and the set of letters occurring infinitely often in t is denoted $\mathrm{alphinf}(t)$. The set $\mathbb{M}(\Sigma, D)$ comprises all finite traces while $\mathbb{R}(\Sigma, D)$ contains all finite or infinite 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 and the empty trace ε is a unit, 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 $[\cdot] : \Sigma^* \to \mathbb{M}(\Sigma, D)$ 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 $[u] = (V, \preceq, \lambda)$ is given by $V = \{i \in \mathbb{N} \mid 0 \leqslant i < |u|\}$, $\lambda(i) = a_i$, and $\preceq = E^*$ with $(i, j) \in E$ iff $i < j$ and $(a_i, a_j) \in D$.

In the following, we are interested in the logic $\mathrm{MSO}(\mathrm{N}, \lessdot)$ that speaks about nodes and sets of nodes of a trace. The logic has *individual* and *set variables*. The syntax of $\mathrm{MSO}(\mathrm{N}, \lessdot)$ is given by

$$\varphi ::= \lambda(x) = a \mid x \lessdot y \mid x = y \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \exists X\, \varphi \mid \exists^{\mathrm{fin}} X\, \varphi$$

where a ranges over N, x, y are individual variables, and X is a set variable.

Formulas of the logic MSO(N, $\lessdot$) will be interpreted over traces. Formally, the semantics is defined for a trace $t = (V, \leqslant, \lambda)$ and an assignment σ that maps first order variables to positions in V and set variables to subsets of V by:

$$
\begin{array}{lll}
t, \sigma \models \lambda(x) = a & \text{if} & \lambda(\sigma(x)) = a \\
t, \sigma \models x \lessdot y & \text{if} & \sigma(x) \lessdot \sigma(y) \\
t, \sigma \models x = y & \text{if} & \sigma(x) = \sigma(y) \\
t, \sigma \models x \in X & \text{if} & \sigma(x) \in \sigma(X) \\
t, \sigma \models \neg\varphi & \text{if} & t, \sigma \not\models \varphi \\
t, \sigma \models \varphi \vee \psi & \text{if} & t, \sigma \models \varphi \text{ or } t, \sigma \models \psi \\
t, \sigma \models \exists x\, \varphi & \text{if} & \text{there exists } v \in V \text{ such that } t, \sigma[x \mapsto v] \models \varphi \\
t, \sigma \models \exists X\, \varphi & \text{if} & \text{there exists } U \subseteq V \text{ such that } t, \sigma[X \mapsto U] \models \varphi \\
t, \sigma \models \exists^{\text{fin}} X\, \varphi & \text{if} & \text{there exists } U \subseteq V \text{ with } U \text{ finite and } t, \sigma[X \mapsto U] \models \varphi
\end{array}
$$

where $\sigma[x \mapsto v]$ is the assignment that differs from σ only in the value of x that now equals v, and similarly for $\sigma[X \mapsto U]$. To make formulas more readable, we will freely use abbreviations such as $\alpha \wedge \beta$, $X \subseteq Y$, $X \cap Y \neq \emptyset$, ... whose obvious intended meaning can easily be expressed by formulas from MSO(N, $\lessdot$).

Sometimes, we write $\varphi(X_1, \ldots, X_k, x_1, \ldots, x_\ell)$ to stress the fact that the *free* variables in φ are among $\{X_1, \ldots, X_k, x_1, \ldots, x_\ell\}$. In this case, we may also write $t \models \varphi(U_1, \ldots, U_k, v_1, \ldots, v_\ell)$ instead of $t, \sigma \models \varphi$ where σ is an assignment satisfying $\sigma(X_i) = U_i \subseteq V$ for $1 \leqslant i \leqslant k$ and $\sigma(x_j) = v_j \in V$ for $1 \leqslant j \leqslant \ell$.

Usually, an MSO-logic over partial orders is defined with the atomic proposition $x \leqslant y$ instead of $x \lessdot y$ and $x = y$. Clearly, the formulas $x \lessdot y$ and $x = y$ can be expressed by first-order formulas using the partial order $\leqslant$, only. Conversely, $\leqslant$ is the reflexive and transitive closure of $\lessdot$, i.e., $x \leqslant y$ is equivalent to

$$\forall X\, [(y \in X \wedge \forall y_1, y_2\, (y_1 \lessdot y_2 \wedge y_2 \in X \to y_1 \in X)) \to x \in X].$$

Thus, using $x \leqslant y$ instead of $x \lessdot y$ and $x = y$ does not change the expressive power of the logic. We have chosen not to include the atomic proposition $x \leqslant y$ directly in the syntax of MSO(N, $\lessdot$) since our upper bound proof relies on the fact that the number of nodes y that are directly related with a fixed node x is bounded by some value which does not depend on the trace but depends on the dependence alphabet only. This would not be the case if we included $\leq$ since the number of nodes dominated by a node x is arbitrary large.

Example 2.1. Consider the following two formulas:

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

of MSO(N, $\lessdot$). Let $t = (V, \leq, \lambda)$ be any trace and $u \in V$ a vertex. Then, for any subset $U \subseteq V$ we have $t \models \text{upset}(u, U)$ iff $U = \uparrow u$ where $\uparrow u = \{v \in V \mid u \leqslant v\}$. Similarly, for any *finite* subset $U \subseteq V$, we have $t \models \text{downset}(u, U)$ iff $U = \downarrow u = \{v \in V \mid v \leqslant u\}$.

Indeed, it is easy to see that $t \models \text{upset}(u, \uparrow u)$. Conversely, assume that $t \models \text{upset}(u, U)$. For $v \in \uparrow u$, an easy induction on the length of a shortest $\lessdot$-path from u to v shows that $v \in U$, i.e., $\uparrow u \subseteq U$. For the converse inclusion, assume that $U \setminus \uparrow u \neq \emptyset$ and let v be minimal in $U \setminus \uparrow u$ (since $\downarrow w$ is finite for any $w \in V$, such a minimal node exists). Then, $v \neq u$ and since $t \models \text{upset}(u, U)$ we find $w \in U$ with $w \lessdot v$. Since v was chosen minimal, we get $w \in \uparrow u$. Hence, $v \in \uparrow u$, a contradiction. Therefore, we obtain $U = \uparrow u$.

Now consider the formula downset(x, X) that is just the dual of upset(x, X) and let U be finite. Then also the proof is dual to the one above. When one shows $U \subseteq \downarrow u$, one assumes $v \in U \setminus \downarrow u$ maximal. Such a maximal node exists since U is finite.

In the following, we will write $X = \downarrow x$ and $X = \uparrow x$ as a more intuitive abbreviation for the formulas downset(x, X) and upset(x, X).

Definition 2.2. An MSO(N, $\lessdot$)-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 2.3. An MSO(N, $\lessdot$)-definable temporal logic is given by

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

Then the syntax of the temporal logic TL(B) is defined by the grammar

$$\varphi ::= M(\underbrace{\varphi, \ldots, \varphi}_{\text{arity}(M)}) \mid a$$

where M ranges over B and a over N.

Let $t = (V, \leq, \lambda)$ be a trace over some finite dependence alphabet (Σ, D) and $\varphi \in \text{TL}(B)$ a formula of 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 $\varphi = a \in \text{N}$, then $\varphi^t = \{v \in V \mid \lambda(v) = a\}$. If $\varphi = M(\varphi_1, \ldots, \varphi_m)$ where $M \in B$ is of arity $m \geqslant 0$, then

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

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

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

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

Example 2.4. First, the boolean connectives negation and conjunction can be expressed by $[\![\neg]\!](X_1, x) = \neg(x \in X_1)$ and $[\![\vee]\!](X_1, X_2, x) = (x \in X_1) \vee (x \in X_2)$.

Existential next: $\mathsf{EX}\,\varphi$ is one of the simplest temporal modalities. Intuitively, $\mathsf{EX}\,\varphi$ means that there is an immediate successor of the current vertex where φ holds. Formally, we can set $[\![\mathsf{EX}]\!](X_1, x) = \exists y\ (x \lessdot y \wedge y \in X_1)$ which is even a first-order formula since it does not use set quantifications.

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

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

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. Since the partial order $\leqslant$ cannot be used directly, we cannot write a first-order formula for the semantics of SU. Instead, the semantics $[\![\mathsf{SU}]\!](X_1, X_2, x)$ can be written as an existential formula:

$$\begin{aligned}\exists X \exists^{\text{fin}} Y \exists y\ (X = {\uparrow}x \wedge Y = {\downarrow}y \wedge y \in X \cap X_2 \\ \wedge\ \forall z (z \in X \cap Y \backslash \{x, y\} \to z \in X_1)).\end{aligned}$$

The classical non-strict version of universal until is $\varphi\ \mathsf{U}\ \psi = \psi \vee (\varphi \wedge (\varphi\ \mathsf{SU}\ \psi))$. Note also that $\mathsf{EX}\,\varphi = \texttt{false}\ \mathsf{SU}\ \varphi$.

Existential until: $\varphi\ \mathsf{EU}\ \psi$ is another binary modality. It claims the existence of some finite path $x_0 \lessdot x_1 \cdots \lessdot x_n$ starting at the current node x_0 and such that ψ holds at x_n and φ holds at x_i for all $0 \leqslant i < n$. Formally, we can define this modality by

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

Existential globally. The formula $\mathsf{EG}\,\varphi$ claims the existence of a *maximal* $\lessdot$-path in the trace, starting from the current vertex, where φ always holds. The corresponding modality can be defined similarly to $[\![\mathsf{EU}]\!]$ by

$$\begin{aligned}[\![\mathsf{EG}]\!](X_1, x) = \exists P\ (x \in P \wedge P \subseteq X_1 \wedge \\ \forall z\ (z \in P \to\ (z = x \vee \exists y\ (y \in P \wedge y \lessdot z)) \\ \wedge\ (\exists y\ (y \in P \wedge z \lessdot y) \vee \neg \exists y\ (z \lessdot y)))).\end{aligned}$$

For more examples, see [10] where most modalities met in the literature on local temporal logics for traces are expressed in terms of MSO(N, $\leq$)-modalities. As $\leq$ can be expressed using $\lessdot$, any of those formulas can be transformed into an equivalent one from MSO(N, $\lessdot$).

In [10, Theorem 9], we showed that the following problem belongs to **PSPACE** where the size $|\varphi|$ of a temporal formula φ is the number of its subformulas.[2]

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

By the above discussion, any MSO(N, $<$)-definable temporal logic is also MSO(N, $\leq$)-definable, i.e., the **PSPACE** upper bound holds for these logics as well. In this paper, we will also consider the finite dependence alphabet as part of the input, i.e., we study the complexity of the following problem:

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

Analyzing the proof of [10, Theorem 9], one obtains the following:

Theorem 2.5 (cf. [10]). *For any* MSO(N, $<$)*-definable temporal logic, the uniform satisfiability problem is elementarily decidable.*

Proof. The proof of [10, Theorem 9] is based on computing automata from the MSO-descriptions of the modalities in a preprocessing step. These computations depend elementarily on the dependence alphabet. Hence, the remaining procedure from [10] can be applied and yields the result. □

For temporal logics based on the classical modalities from Example 2.4 as well as on Thiagarajan's process-based modalities from [22], we solved the uniform satisfiability problem in **PSPACE** [8]. In this paper, we present matching lower and upper bounds for the uniform satisfiability problem of arbitrary MSO(N, $<$)-definable temporal logics. These bounds are expressed in terms of the number of monadic quantifier alternations in the formulas $[\![M]\!]$. Following [6], $\mathrm{M}\Sigma_n^1(\mathrm{N}, <)$ comprises all MSO(N, $<$)-formulae that are logically equivalent to one of the form $\exists \overrightarrow{X_1} \forall \overrightarrow{X_2} \ldots \exists/\forall \overrightarrow{X_n} \varphi$ where φ does not contain any second-order quantification. Here, $\overrightarrow{Y}$ stands for a tuple of set variables. For instance, all modalities from Example 2.4 have been defined by $\mathrm{M}\Sigma_1^1(\mathrm{N}, <)$-formulae.

Dually, a formula belongs to $\mathrm{M}\Pi_n^1(\mathrm{N}, <)$ if and only if its negation is an element of $\mathrm{M}\Sigma_n^1(\mathrm{N}, <)$. Finally, $\mathrm{Bool}\mathrm{M}\Sigma_n^1(\mathrm{N}, <)$ is the set of Boolean combinations of formulas from $\mathrm{M}\Sigma_n^1(\mathrm{N}, <)$. We write $\mathrm{FO}(\mathrm{N}, <)$ for $\mathrm{M}\Sigma_0^1(\mathrm{N}, <) = \mathrm{M}\Pi_0^1(\mathrm{N}, <)$, i.e., for those formulas that can be written without set quantification. When $\mathcal{L}$ is a logic such as $\mathrm{FO}(\mathrm{N}, <)$, $\mathrm{M}\Pi_n^1(\mathrm{N}, <)$, ..., we speak of an $\mathcal{L}$-modality M if $[\![M]\!] \in \mathcal{L}$, and of an $\mathcal{L}$-definable temporal logic TL(B) whenever all modalities are $\mathcal{L}$-modalities.

Remark 2.6. Let TL(B) be some $\mathrm{Bool}\mathrm{M}\Sigma_n^1(\mathrm{N}, <)$-definable temporal logic. Then there is a finite set H of $\mathrm{M}\Sigma_n^1(\mathrm{N}, <)$-modalities such that, for every $M \in B$, $[\![M]\!]$ is a Boolean combination of formulas from H. In addition, we can assume $\neg, \vee \in H$. Now let φ be a TL(B)-formula. Replacing every occurrence in φ of a modality $M \in B$ with the equivalent Boolean combination of formulas from H yields an equivalent formula ψ from TL(H). Recall that the size of φ is the number of its subformulas; hence $|\psi|$ is linear in $|\varphi|$, i.e., we reduced the uniform satisfiability problem for the $\mathrm{Bool}\mathrm{M}\Sigma_n^1(\mathrm{N}, <)$-definable temporal logic TL(B) to that of the $\mathrm{M}\Sigma_n^1(\mathrm{N}, <)$-definable temporal logic TL(H).

As a consequence, it will suffice to prove the upper complexity bound for $\mathrm{M}\Sigma_n^1(\mathrm{N}, <)$-definable temporal logics. Dually, the lower bound will be proved for $\mathrm{M}\Pi_n^1(\mathrm{N}, <)$-definable temporal logics, only. From the same reduction, we obtain that it holds for $\mathrm{M}\Sigma_n^1(\mathrm{N}, <)$-definable temporal logics as well.

3. n-EXPSPACE upper bound for $\mathrm{M}\Sigma_n^1$-logics

It is the aim of this section to prove an upper bound for the uniform satisfiability problem sharper than that given in Theorem 2.5. To state this upper bound, let $\mathtt{poly}(n)$ denote the set of polynomial functions of one argument. The function $\mathtt{tower} : \mathbb{N}^2 \to \mathbb{N}$ is defined inductively by $\mathtt{tower}(0, m) = m$ and by $\mathtt{tower}(\ell, m) = 2^{\mathtt{tower}(\ell-1,m)}$ for $\ell > 0$. Now we can state the main result of this section:

Theorem 3.1. *Let* TL *be some* $\mathrm{M}\Sigma_n^1(\mathrm{N}, <)$*-definable temporal logic. Then the uniform satisfiability problem for* TL *can be solved in space* $\mathtt{poly}(|\varphi|) \cdot \mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$*, i.e., it is in* **n-EXPSPACE**.

Remark 3.2. To avoid unnecessary complications, we give the proof for infinite traces only. Hence, we use Büchi automata over ω-words representing infinite traces. We can also deal with finite traces similarly, using automata over finite words. This is left to the reader.

[2] In [10], we did not allow the restriction of set quantification to finite sets in the modalities $[\![M]\!]$, but the necessary additions are obvious.

3.1. The decision procedure – Proof of Theorem 3.1

The decision procedure we propose refines ideas from [10] that can also be found (although in a different presentation) in [14]. The main ingredient of the decision procedure are *modality automata* defined below. Let $w = a_0a_1 \cdots \in \Sigma^\omega$ be a word over Σ and $X_i \subseteq \mathbb{N}$ be sets for $1 \leqslant i \leqslant m$. Then $(w, X_1, \ldots, X_m)$ denotes the word $b_0b_1 \ldots$ over $\Sigma \times \{0,1\}^m$ with $b_i = (a_i, x_i^1, x_i^2, \ldots, x_i^m)$ and $x_i^j = 1$ iff $i \in X_j$.

Definition 3.3. Let (Σ, D) be a finite dependence alphabet and α an m-ary MSO(N, $\lessdot$)-modality. A Büchi-automaton $\mathcal{A}$ over $\Sigma \times \{0,1\}^{m+1}$ is called *modality automaton for α over* (Σ, D) if a word $(w, X_0, X_1, \ldots, X_m)$ is accepted by $\mathcal{A}$ iff $[w] \models \forall x\, (x \in X_0 \leftrightarrow \alpha(X_1, X_2, \ldots, X_m, x))$ where $[w]$ is the trace induced by w.

Our decision procedure will have to construct modality automata for all $\mathrm{M}\Sigma_n^1(\mathrm{N}, \lessdot)$-modalities α in the temporal logic. The modality automaton $\mathcal{A}_\alpha$ for α over (Σ, D) is a Büchi automaton for the $\mathrm{M}\Pi_{n+1}^1(\mathrm{N}, \lessdot)$-formula

$$\alpha' = \forall x\, (x \in X_0 \leftrightarrow \alpha(X_1, \ldots, X_m, x)).$$

In α we find atomic propositions of the form $y \lessdot z$. Reading a word w we can check whether two positions $i, j < |w|$ are consecutive (i.e., satisfy $i \lessdot j$) in the trace $[w]$ by keeping a subset of Σ in the state. Then, using classical constructions on automata (projection for existential quantification, complement for negation and disjoint union for disjunction) we can construct a modality automaton for α over (Σ, D). Note that a universal quantification $\forall = \neg\exists\neg$ needs two complements and yields two exponentials. Hence, this naïve approach yields an exponential tower whose height is the number of quantifier alternations in α' (including first-order quantifiers), even for FO(N, $\lessdot$)-modalities. Since the space bound in Theorem 3.1 mentions only alternations of set quantifiers, we need the following more efficient construction.

Proposition 3.4. *Let $n \geqslant 1$ and α be an $\mathrm{M}\Sigma_n^1(\mathrm{N}, \lessdot)$-modality. Then the following problem can be solved in space* $\mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$

input: *a finite dependence alphabet* (Σ, D)
output: *a modality automaton for α over* (Σ, D).

The proof of this proposition will be presented in Section 3.4 and use the concepts and results from Sections 3.2 and 3.3.

Before we explain how to use modality automata to solve the uniform satisfiability problem, we fix some more notation. Let φ and ξ be TL(B)-formulas. Then topM(ξ) denotes the outermost modality name of ξ. We write $\xi \leq \varphi$ if ξ is a subformula of φ (this includes the case $\varphi = \xi$). Furthermore, $\mathrm{Sub}(\varphi) = \{\xi \in \mathrm{TL}(B) \mid \xi \leq \varphi\}$ is the set of subformulas of φ. For an alphabet Σ, we will consider words of the form $(w, (X_\xi)_{\xi \leqslant \varphi})$ with $w \in \Sigma^\omega$ and $X_\xi \subseteq \mathbb{N}$, i.e., words over the extended alphabet $\Sigma_\varphi = \Sigma \times \{0,1\}^{\mathrm{Sub}(\varphi)}$. For a subformula $\xi = M(\xi_1, \ldots, \xi_m) \leq \varphi$ and a letter $\overline{a} = (a, (x_\xi)_{\xi \leq \varphi}) \in \Sigma_\varphi$, let $\overline{a}{\restriction}\xi = (a, x_\xi, x_{\xi_1}, \ldots, x_{\xi_m})$. Similarly, for a word $\overline{w} = (w, (X_\xi)_{\xi \leq \varphi}) \in \Sigma_\xi^\omega$, let $\overline{w}{\restriction}\xi = (w, X_\xi, X_{\xi_1}, \ldots, X_{\xi_m})$.

Recall the following decision procedure from [8] that we repeat here for the sake of completeness. Let φ be some TL(B)-formula and (Σ, D) some finite dependence alphabet. Furthermore, suppose that for each $M \in B$, we are given a modality automaton $\mathcal{A}_M$ for $[\![M]\!]$ over (Σ, D) with set of states Q_M. From these modality automata, we construct an automaton $\mathcal{A}$ over Σ_φ. The set of states is $Q = \prod_{\xi \leq \varphi} Q_{\mathrm{topM}(\xi)}$. For a letter $\overline{a} \in \Sigma_\varphi$ and states $p = (p_\xi)_{\xi \leq \varphi}$ and $q = (q_\xi)_{\xi \leq \varphi}$, we have a transition $p \xrightarrow{\overline{a}} q$ in $\mathcal{A}$ if and only if, for all $\xi \leq \varphi$, we have $p_\xi \xrightarrow{\overline{a}{\restriction}\xi} q_\xi$ in the modality automaton $\mathcal{A}_{\mathrm{topM}(\xi)}$. With this definition, a sequence of states $p^0, p^1, \ldots$ is a run of $\mathcal{A}$ on a word $\overline{w} = (w, (X_\xi)_{\xi \leq \varphi}) \in \Sigma_\varphi^\omega$ if and only if for each $\xi \leqslant \varphi$, its projection $p_\xi^0, p_\xi^1, \ldots$ on ξ is a run of the modality automaton $\mathcal{A}_{\mathrm{topM}(\xi)}$ for the word $\overline{w}{\restriction}\xi$. A run $p^0, p^1, \ldots$ of $\mathcal{A}$ is accepting if and only if for each $\xi \leqslant \varphi$, its projection $p_\xi^0, p_\xi^1, \ldots$ is accepting in the modality automaton $\mathcal{A}_{\mathrm{topM}(\xi)}$. So $\mathcal{A}$ is a generalized Büchi automaton which has the following useful property:

Proposition 3.5 ([8, Proposition 4.1])**.** *The formula $\varphi \in \mathrm{TL}(B)$ is satisfiable by some trace over (Σ, D) if and only if $\mathcal{A}$ accepts some word $\overline{w} = (w, (X_\xi)_{\xi \leqslant \varphi}) \in \Sigma_\varphi$ with $X_\varphi \neq \emptyset$.*

Sketch of proof. Let $\overline{w} = (w, (X_\xi)_{\xi \leqslant \varphi}) \in \Sigma_\varphi^\omega$. Then one shows that $\overline{w}$ is accepted by $\mathcal{A}$ if and only if for each $\xi \leqslant \varphi$ we have $X_\xi = \xi^{[w]} = \{p \in \mathbb{N} \mid [w], p \models \xi\}$ [8, Lemma 4.1]. This immediately implies the statement.

Proof of Theorem 3.1. The satisfiability of φ is (essentially) equivalent to the non-emptiness problem for the automaton $\mathcal{A}$. To solve it non-deterministically, we only need to keep in memory a few $|\varphi|$-tuples of states of our modality automata, and some counter (counting up to $|\varphi|$) to check the generalized Büchi condition. By Proposition 3.4, modality automata can be computed in space $\mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$. Hence, the transition relation of the automaton $\mathcal{A}$ can be decided in space $\mathtt{poly}(|\varphi|) \cdot \mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$. Thus, its non-emptiness can be decided in space $\mathtt{poly}(|\varphi|) \cdot \mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$ which finishes the proof of Theorem 3.1. □

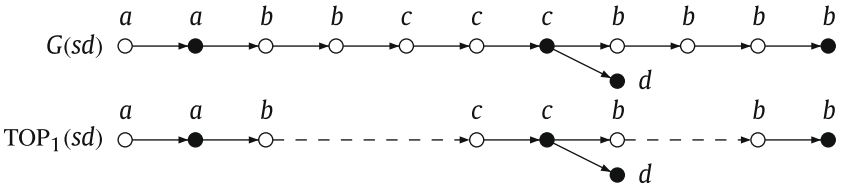

Fig. 1. Update of $\mathrm{TOP}_1(s)$.

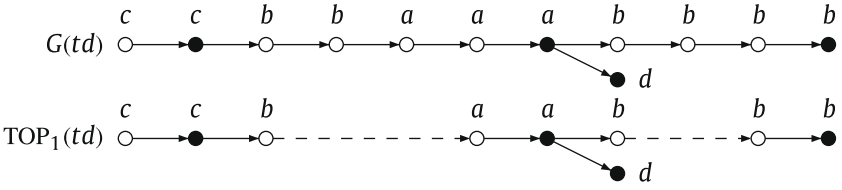

Fig. 2. Update of $\mathrm{TOP}_1(t)$.

The still missing proof of Proposition 3.4 will be given in Section 3.4. It relies on a locality theorem by Schwentick and Bartelmann [21] (cf. Proposition 3.15). In essence, it says that an FO(N, $\lessdot$)-formula is effectively equivalent to the possibility of placing some pebbles such that any sphere in the structure extended by these pebbles satisfies some first-order property. Therefore, the following section defines spheres in traces and shows how they can be computed by an automaton.

3.2. Spheres

Throughout this section, fix some dependence alphabet (Σ, D) and write $\mathbb{M}$ for $\mathbb{M}(\Sigma, D)$ and similarly $\mathbb{R}$ for $\mathbb{R}(\Sigma, D)$. The *trace graph* of a trace $t = (V, \leq, \lambda)$ is the structure $G(t) = (V, \leq, \lessdot, \lambda)$. The *restriction* of a structure $\mathcal{M} = (V, \leq, \lessdot, \lambda)$ to $U \subseteq V$ is the structure

$$\mathcal{M}{\restriction}U = (U, \leq \cap U^2, \lessdot \cap U^2, \lambda{\restriction}U).$$

If $\mathcal{M} = G(t)$ is a trace graph, $\mathcal{M}{\restriction}U$ need not be a trace graph itself. In particular, the relation $\lessdot \cap U^2$ need not be the covering relation of $\leq \cap U^2$. We let $\bowtie = (\lessdot \cup \gtrdot)$. A *path of length n* in $\mathcal{M}$ is a sequence $x_0 \bowtie x_1 \ldots \bowtie x_n$ with $x_i \in V$, i.e., consecutive elements are related by $\lessdot$ in any direction. For $x, y \in V$, the *distance* $d_{\mathcal{M}}(x, y)$ is the minimal length of a path from x to y. The distance is generalized to $x \in V$ and $U \subseteq V$ by $d_{\mathcal{M}}(x, U) = \min\{d_{\mathcal{M}}(x, y) \mid y \in U\}$. For $r \in \mathbb{N}$ and $U \subseteq V$, let $\mathrm{sph}_r(\mathcal{M}, U) = \{x \in V \mid d_{\mathcal{M}}(x, U) \leqslant r\}$ consist of all elements of V whose distance to U is at most r. Then the *sphere* $\mathrm{SPH}_r(\mathcal{M}, U)$ *around* U denotes the substructure $\mathcal{M}{\restriction}\mathrm{sph}_r(\mathcal{M}, U)$.

Let $t = (V, \leq, \lambda)$ be a finite trace. 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 $\mathrm{last}(t) = \{\mathrm{last}_a(t) \mid a \in \mathrm{alph}(t)\}$. Then $\mathrm{top}_r(t)$ denotes $\mathrm{sph}_r(t, \mathrm{last}(t))$ and $\mathrm{TOP}_r(t) = \mathrm{SPH}_r(t, \mathrm{last}_r(t))$. Hence, $\mathrm{top}_r(t) = \{x \in V \mid d_{G(t)}(x, \mathrm{last}(t)) \leqslant r\}$ and $\mathrm{TOP}_r(t) = G(t){\restriction}\mathrm{top}_r(t)$. We will first show in Lemma 3.7 that the top spheres can be computed incrementally reading an arbitrary linearization of a trace.

Example 3.6. Let $\Sigma = \{a, b, c, d\}$ with independence relation $I = \{(b, d), (d, b), (a, c), (c, a)\}$ and consider the trace $s = [aabbcccbbbb]$. In Fig. 1, the trace graph of sd 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 line, 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 $\lessdot$ 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 $\leqslant$ in the top sphere. Lemma 3.7 shows that this information is sufficient to compute $\mathrm{TOP}_r(sd)$ from $\mathrm{TOP}_r(s)$ and d.

Lemma 3.7. *Let s be a finite trace, $a \in \Sigma$, and $r \in \mathbb{N}$. Then $\mathrm{TOP}_r(sa)$ can be computed from $\mathrm{TOP}_r(s)$ and the letter a in time polynomial in $|\mathrm{TOP}_r(s)| + |\Sigma|$.*

Proof. Let $G(sa) = (V \uplus \{x\}, \leqslant, \lessdot, \lambda)$ with $x = \mathrm{last}_a(sa)$. Note that $G(s) = G(sa){\restriction}V$. We first show that for $y \in V$, we have $y \lessdot x$ if and only if $y \in \mathrm{last}(s)$ and $\lambda(y) \in D(a)$ and $y < z$ implies $\lambda(z) \in I(a)$ for all $z \in \mathrm{last}(s)$. Note that this necessary and sufficient condition for $y \lessdot x$ can be checked using $\mathrm{TOP}_r(s)$ and the letter a, only.

Assume $y \lessdot x$ and let $b = \lambda(y)$. Then, $(a, b) \in D$ and $y \leqslant \mathrm{last}_b(s) < x$. We deduce that $y = \mathrm{last}_b(s)$. Now, if $y < z$ then $z \not\leqslant x$ and it follows $\lambda(z) \in I(a)$. Conversely, let $y \in \mathrm{last}(s)$ with $\lambda(y) \in D(a)$ such that $y < z$ implies $\lambda(z) \in I(a)$ for all

$z \in \text{last}(s)$. We have $y < x$ because of $(\lambda(y), a) \in D$. Now, let z be such that $y \leqslant z \lessdot x$. Then, $c = \lambda(z) \in D(a)$ and since $z \leqslant \text{last}_c(s) < x$ we get $z = \text{last}_c(s) \in \text{last}(s)$. We deduce that $y = z$ and $y \lessdot x$.

Next we prove that a vertex is in $\text{top}_r(sa)$ if it is either x, or its distance from some $y \lessdot x$ is at most $r-1$, or its distance from some $\text{last}_b(s)$ for $b \neq a$ is at most r, i.e.,

$$\begin{aligned}\text{top}_r(sa) = \{x\} &\cup \text{sph}_{r-1}(\text{TOP}_r(s), \{y \mid y \lessdot x\}) \\ &\cup \text{sph}_r(\text{TOP}_r(s), \{\text{last}_b(s) \mid b \in \text{alph}(s)\backslash\{a\}\}).\end{aligned}$$

Note again that this set can be computed from $\text{TOP}_r(s)$ and a.

The inclusion $\supseteq$ is clear. Conversely, let $y \in \text{top}_r(sa)\backslash\{x\}$ and let $y = y_0 \bowtie \cdots \bowtie y_p \in \text{last}(sa)$ be a shortest path from y to $\text{last}(sa)$. We have $p \leqslant r$. If $y_p = x$ then $p > 0$, the path $y = y_0 \bowtie \cdots \bowtie y_{p-1}$ is in $\text{TOP}_r(s)$ and $y_{p-1} \lessdot x$. Therefore, $y \in \text{sph}_{r-1}(\text{TOP}_r(s), \{y \mid y \lessdot x\})$. If $y_p \neq x$ then the whole path $y = y_0 \bowtie \cdots \bowtie y_p$ is in $\text{TOP}_r(s)$, $y_p \in \text{last}(s)$ and $\lambda(y_p) \neq a$. Therefore, $y \in \text{sph}_r(\text{TOP}_r(s), \{\text{last}_b(s) \mid b \in \text{alph}(s)\backslash\{a\}\})$.

Finally, it remains to show that the relations $\leqslant^{sa}$ and $\lessdot^{sa}$ in $\text{TOP}_r(sa)$ can also be computed from the relations $\leqslant^s$ and $\lessdot^s$ in $\text{TOP}_r(s)$. Again, we use the fact that we know how to decide $y \lessdot x$ from $\text{TOP}_r(s)$ and a. For $y, z \in \text{top}_r(sa)$, we have

- $y \lessdot^{sa} z$ if and only if $y, z \in \text{top}_r(s)$ and $y \lessdot^s z$ or $y \in \text{top}_r(s)$, $z = x$ and $y \leqslant^s y' \lessdot x$ for some $y' \in \text{last}(s)$.
- $y \leqslant^{sa} z$ if and only if $y, z \in \text{top}_r(s)$ and $y \leqslant^s z$ or $y \lessdot x = z$. □

Let $w = a_0a_1a_2 \cdots \in \Sigma^\omega$ and $t = [w] = (V, \preceq, \lambda) \in \mathbb{R}(\Sigma, D)$ with $V = \mathbb{N}$. Fix also some $r \in \mathbb{N}$. A modality automaton will have to check properties of spheres of the form $\text{SPH}_r(G(t), x)$. For each $x \in V$, we can find a finite prefix u of w such that $\text{SPH}_r(G(t), x)$ is contained in $\text{TOP}_{2r}([u])$. Indeed, let j be minimal with $\text{sph}_r(G(t), x) \subseteq \{0, \ldots, j\}$ and let $u = a_0 \cdots a_j$ be the corresponding finite prefix of w. We have $j \in \text{last}([u])$ and $d_{G(t)}(x, j) \leqslant r$, hence also $d_{G([u])}(x, j) \leqslant r$. Therefore, $\text{sph}_r(G(t), x) \subseteq \text{top}_{2r}([u])$.

From Lemma 3.7, the structures $\text{TOP}_{2r}([u])$ for all finite prefixes of w can be computed by an automaton. But, in order to check properties of spheres, we also need to determine when a vertex x in $\text{TOP}_{2r}([u])$ is such that $\text{SPH}_r(G(t), x)$ is contained in $\text{TOP}_{2r}([u])$. This is the purpose of the following definitions and lemmas. We give two sufficient conditions (r-critical and r-safe) ensuring the above containment. The first one requires that the distance from x to $\text{last}([u])$ is r and will increase when we add a new letter to the prefix u of w.

Definition 3.8. Let $s \in \mathbb{M}$ be a finite trace, x be a vertex in s and $a \in \Sigma$. Then, x is *r-critical* for (s, a) if $x \in \text{top}_r(s)$ but $x \notin \text{top}_r(sa)$.

Note that we can determine in polynomial time whether a vertex $x \in \text{top}_r(s)$ is r-critical for (s, a) just knowing $\text{TOP}_r(s)$ and a since by Lemma 3.7, we can determine $\text{top}_r(sa)$ from $\text{TOP}_r(s)$ and a.

Lemma 3.9. *Let $s \in \mathbb{M}$ be a finite trace, x be a vertex in s and $a \in \Sigma$. If x is r-critical for (s, a) then for all $t \in \mathbb{R}$, we have* $\text{SPH}_r(sat, x) = \text{SPH}_r(\text{TOP}_{2r}(s), x)$.

Proof. We first show that $\text{sph}_r(sat, x) \subseteq \text{top}_{2r}(s)$. Let $y \in \text{sph}_r(sat, x)$ and let $x = x_0 \bowtie \cdots \bowtie x_n = y$ be a shortest path in $G(sat)$ from x to y. We have $n \leqslant r$. We show by contradiction that this path must be in $G(s)$. So assume that this is not the case and consider the least k with x_k in $G(s)$ and x_{k+1} not in $G(s)$. We must have $x_k \lessdot x_{k+1}$. With $b = \lambda(x_k)$, we deduce that $x_k \leqslant \text{last}_b(s) < x_{k+1}$ and by definition of $\lessdot$ we get $x_k = \text{last}_b(s)$. Now, if $b \neq a$ then $x_k \in \text{last}(sa)$ and $d_{G(sa)}(x, x_k) \leqslant k < r$ since the path $x = x_0 \bowtie \cdots \bowtie x_k$ is in $G(s)$, hence also in $G(sa)$. This is a contradiction with x being r-critical for (s, a). Assume now $b = a$. Then $x_k \lessdot x_{k+1}$ implies $(a, \lambda(x_{k+1})) \in D$ and $x_{k+1} \in G(sat)\backslash G(s)$ implies $x_{k+1} \neq \text{last}_a(sa)$. Together, we obtain $\text{last}_a(sa) \leqslant x_{k+1}$. From $x_k \in G(s)$ and $\lambda(x_k) = a$, we infer $x_k < \text{last}_a(sa) \leqslant x_{k+1}$ and using $x_k \lessdot x_{k+1}$ again we deduce $x_{k+1} = \text{last}_a(sa)$. But then $d_{G(sa)}(x, x_{k+1}) \leqslant k + 1 \leqslant r$, which is again a contradiction with x being r-critical for (s, a). Therefore, the whole path $x = x_0 \bowtie \cdots \bowtie x_n = y$ is in $G(s)$ and $d_{G(s)}(x, y) = n \leqslant r$. Since x is r-critical for (s, a) we have $x \in \text{top}_r(s)$ and we deduce that $y \in \text{top}_{2r}(s)$ as desired.

Next, for $y, z \in \text{sph}_r(sat, x)$ we have $y \leqslant z$ (respectively, $y \lessdot z$) in $\text{SPH}_r(sat, x)$ iff the same holds in $G(sat)$ iff the same holds in $G(s)$ iff the same holds in $\text{TOP}_{2r}(s)$. Therefore, $\text{SPH}_r(sat, x) = \text{SPH}_r(\text{TOP}_{2r}(s), x)$. □

The above condition deals with vertices that will eventually leave the top r-sphere. But there are vertices that may stay forever in the top r-sphere. This fact depends on the alphabet B of the trace that remains to be read.

Definition 3.10. Let $s \in \mathbb{M}$ be a finite trace, x be a vertex in s and $B \subseteq \Sigma$. Then, x is *r-safe* for (s, B) if $x \in \text{top}_r(s)$, $B \subseteq \text{alph}(s)$ and for all $b \in B$, $d_{G(s)}(x, \text{last}_b(s)) > r$ and if $\text{last}_b(s) \leqslant \text{last}_a(s)$ for some $a \in \text{alph}(s)$ then $a \in B$.

Note that we can determine in polynomial time whether a vertex $x \in \text{top}_r(s)$ is r-safe for (s, B) just knowing $\text{TOP}_r(s)$ and B.

Lemma 3.11. *Let $s \in \mathbb{M}$ be a finite trace, x be a vertex in s and $B \subseteq \Sigma$. If x is r-safe for (s, B) then for all $t \in \mathbb{R}$ such that $\mathrm{alph}(t) \subseteq B$, we have $\mathrm{SPH}_r(st, x) = \mathrm{SPH}_r(\mathrm{TOP}_{2r}(s), x)$.*

Proof. As in the proof of Lemma 3.9, it is enough to show that $\mathrm{sph}_r(st, x) \subseteq \mathrm{top}_{2r}(s)$. Let $y \in \mathrm{sph}_r(st, x)$ and let $x = x_0 \bowtie \cdots \bowtie x_n = y$ be a shortest path in $G(st)$ from x to y. We have $n \leqslant r$. We show by contradiction that this path must be in $G(s)$. First note that this will conclude the proof since in this case we get $d_{G(s)}(x, y) = n \leqslant r$ and using $x \in \mathrm{top}_r(s)$ we obtain $y \in \mathrm{top}_{2r}(s)$.

So assume that the path is not in $G(s)$ and consider the least k with x_k in $G(s)$ and x_{k+1} not in $G(s)$. Then, the path $x = x_0 \bowtie \cdots \bowtie x_k$ is in $G(s)$, $k < r$, and $x_k \lessdot x_{k+1}$. With $a = \lambda(x_k)$, we deduce that $x_k \leqslant \mathrm{last}_a(s) < x_{k+1}$ and by definition of $\lessdot$ we get $x_k = \mathrm{last}_a(s)$. Hence, $d_{G(s)}(x, \mathrm{last}_a(s)) = k < r$, which implies $a \notin B$ since x is r-safe for (s, B). Now, $b = \lambda(x_{k+1}) \in \mathrm{alph}(t) \subseteq B$. Since $x_k \lessdot x_{k+1}$, we have $(a, b) \in D$ and $\mathrm{last}_a(s)$ and $\mathrm{last}_b(s)$ must be ordered. Since $\mathrm{last}_b(s) < x_{k+1}$ and $\mathrm{last}_a(s) = x_k \lessdot x_{k+1}$, the ordering must be $\mathrm{last}_b(s) \leqslant \mathrm{last}_a(s)$. Using again the fact that x is r-safe for (s, B) we get $a \in B$, a contradiction since we have already obtained $a \notin B$. □

Lemma 3.12. *Let $w \in \Sigma^\omega$ be an infinite word and let x be a vertex in the associated trace $[w]$. Then,*

1. *either we find a factorization $w = uav$ such that x is r-critical for $([u], a)$,*
2. *or we find a factorization $w = uv$ such that x is r-safe for $([u], \mathrm{alph}(v))$.*

Proof. Write $w = a_0a_1a_2\cdots$ and $[w] = (V, \preceq, \lambda)$ with $V = \mathbb{N}$. We have $x \in \mathrm{top}_r([a_0 \cdots a_x])$. If there exists $i \geqslant x$ such that $x \notin \mathrm{top}_r([a_0 \cdots a_{i+1}])$ then take the least such i and let $u = a_0 \cdots a_i$. By definition, we have x is r-critical for $([u], a_{i+1})$ and we are in the first case.

Assume now that $x \in \mathrm{top}_r([a_0 \cdots a_i])$ for all $i \geqslant x$. Let $B = \mathrm{alphinf}(w)$. Since $\mathrm{sph}_r([w], x)$ is finite, we find a factorization $w = u_1u_2v$ with $\mathrm{sph}_r([w], x)$ contained in $[u_1]$ and $\mathrm{alph}(u_2) = \mathrm{alph}(v) = B$. We show that x is r-safe for $([u], B)$ with $u = u_1u_2$. We have $x \in \mathrm{top}_r([u])$ since by hypothesis, this is true of all prefixes extending u_1. Clearly, $B = \mathrm{alph}(u_2) \subseteq \mathrm{alph}([u])$. Now let $b \in B$. The vertex $\mathrm{last}_b([u])$ must be in $[u_2]$ since $\mathrm{alph}(u_2) = B$. We deduce $d_{G([u])}(x, \mathrm{last}_b([u])) > r$ since $\mathrm{sph}_r([w], x)$ is contained in $[u_1]$. Finally, if $\mathrm{last}_b([u]) \leqslant \mathrm{last}_a([u])$ for some $a \in \mathrm{alph}([u])$ then $\mathrm{last}_a([u])$ must be in $[u_2]$ and $a \in \mathrm{alph}([u_2]) = B$. □

3.3. Automata for φ and for $\forall x\, \varphi$ with $\varphi \in \mathrm{M}\Sigma_n^1(\mathrm{N}, \lessdot)$

Recall that a modality automaton for the $\mathrm{M}\Sigma_n^1(\mathrm{N}, \lessdot)$-modality α is an automaton for the formula $\forall x\ (x \in X_0 \leftrightarrow \alpha(X_1, \ldots, X_m, x))$ which can be rewritten into

$$\forall x\ (x \notin X_0 \vee \alpha(X_1, \ldots, X_m, x)) \wedge \neg\exists x\ (\alpha(X_1, \ldots, X_m, x) \wedge x \notin X_0).$$

This is a conjunction of formulas of the form $\forall x\, \varphi$ and $\neg\varphi$ with $\varphi \in \mathrm{M}\Sigma_n^1(\mathrm{N}, \lessdot)$. Therefore, the following two propositions will be beneficial in the construction of modality automata.

Proposition 3.13. *Let $\varphi(X_1, \ldots, X_m)$ be a formula from $\mathrm{M}\Sigma_n^1(\mathrm{N}, \lessdot)$ with $n \geqslant 1$. 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 Büchi-automaton $\mathcal{A}_\varphi$ over $\Sigma \times \{0, 1\}^m$ that accepts precisely the words $(w, X_1, \ldots, X_m)$ with $[w] \models \varphi(X_1, \ldots, X_m)$.*

Proposition 3.14. *Let $\varphi(X_1, \ldots, X_m, x)$ be a formula from $\mathrm{M}\Sigma_n^1(\mathrm{N}, \lessdot)$. Then the following problem can be solved in space $\mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$*

input: *a finite dependence alphabet (Σ, D)*
output: *a Büchi-automaton $\mathcal{B}_\varphi$ over $\Sigma \times \{0, 1\}^m$ that accepts precisely the words $(w, X_1, \ldots, X_m)$ with $[w] \models \forall x\, \varphi(X_1, \ldots, X_m, x)$.*

We first prove Proposition 3.14 using Proposition 3.13.

Proof. For $n = 0$, the formula $\forall x\, \varphi$ belongs to $\mathrm{FO}(\mathrm{N}, \lessdot) \subseteq \mathrm{M}\Sigma_1^1(\mathrm{N}, \lessdot)$, hence the result follows from Proposition 3.13.

Assume now $n \geq 1$. Consider the $\mathrm{M}\Sigma_n^1(\mathrm{N}, \lessdot)$-formula

$$\varphi'(X_1, \ldots, X_{m+1}) = \exists x\ (X_{m+1} = \{x\} \wedge \varphi).$$

From Proposition 3.13, we can construct in space $\mathtt{tower}(n - 1, \mathtt{poly}(|\Sigma|))$ a Büchi-automaton $\mathcal{A}_{\varphi'}$ for φ'. Note that $\forall x\, \varphi(X_1, \ldots, X_m, x)$ is equivalent with $\forall x\, \varphi'(X_1, \ldots, X_m, \{x\})$. Therefore, we have to construct an automaton for the universal language of $\mathcal{A}_{\varphi'}$:

$$\mathcal{L}_\forall(\mathcal{A}_{\varphi'}) = \{(w, X_1, \ldots, X_m) \mid \forall x\, (w, X_1, \ldots, X_m, \{x\}) \in \mathcal{L}(\mathcal{A}_{\varphi'})\}.$$

By [8, Proposition 7.3], we know that given $\mathcal{A}_{\varphi'}$ this problem can be solved in space $\mathcal{O}(|Q| \log |Q|)$ where Q is the set of states of $\mathcal{A}_{\varphi'}$.[3] Since $\mathcal{A}_{\varphi'}$ can be constructed in space $\mathtt{tower}(n-1, \mathtt{poly}(|\Sigma|))$, its number of states is in $\mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$. Therefore, the automaton for $\mathcal{L}_\forall(\mathcal{A}_{\varphi'})$ can be constructed in space

$$\mathtt{poly}(\mathtt{tower}(n, \mathtt{poly}(|\Sigma|))) = \mathtt{tower}(n, \mathtt{poly}(|\Sigma|)). \quad \square$$

The rest of this section is devoted to the proof of Proposition 3.13. Note that φ can be written as

$$\exists^{(\mathrm{fin})} \overrightarrow{Y_1} \forall^{(\mathrm{fin})} \overrightarrow{Y_2} \cdots \forall^{(\mathrm{fin})} / \exists^{(\mathrm{fin})} \overrightarrow{Y_n}\ \beta'\left(X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n}\right)$$

for some formula $\beta' \in \mathrm{FO}(\mathrm{N}, \lessdot)$ where $\exists^{(\mathrm{fin})} \overrightarrow{Y}$ stands for a sequence of quantifications of the form $\exists Y_i$ and $\exists^{\mathrm{fin}} Y_j$ and similarly for $\forall^{(\mathrm{fin})}$.

Using $\forall = \neg\exists\neg$ and $\forall^{\mathrm{fin}} = \neg\exists^{\mathrm{fin}}\neg$, this can further be rewritten as

$$\exists^{(\mathrm{fin})} \overrightarrow{Y_1} \neg\exists^{(\mathrm{fin})} \overrightarrow{Y_2} \neg\exists^{(\mathrm{fin})} \overrightarrow{Y_3} \cdots \neg\exists^{(\mathrm{fin})} \overrightarrow{Y_n}\ \beta\left(X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n}\right)$$

where $\beta = \beta'$ if n is odd and $\beta = \neg\beta'$ if n is even. To simplify the notation, we let $\overrightarrow{Z} = (Z_1, \ldots, Z_p) = (X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n})$. We will show that we can construct an automaton for $\beta(\overrightarrow{Z})$ in space $\mathtt{poly}(|\Sigma|)$. Then, Proposition 3.13 follows easily as shown at the end of this section.

We have $\beta(\overrightarrow{Z}) \in \mathrm{FO}(\mathrm{N}, \overrightarrow{Z}, \lessdot)$. Considering $Z_1, \ldots, Z_p$ as new predicates, we use Schwentick and Bartelmann's locality theorem [21, Theorem 3.3] that allows to reduce first-order formulae to local formulae.[4] A first-order formula γ is *r-local around the variable y* if it is obtained from some first-order formula δ by replacing any subformula of the form $\exists z\, \varphi$ (respectively, $\forall z\, \varphi$) with $\exists z\, (d_{\bowtie}(y, z) < r \wedge \varphi)$ (respectively, $\forall z\, (d_{\bowtie}(y, z) < r \to \varphi)$) where $d_{\bowtie}(y, z) < r$ is an abbreviation for the straightforward $\mathrm{FO}(\lessdot)$ formula which expresses that there is some $\bowtie$-path from y to z of length at most $r-1$ (recall that $\bowtie = \lessdot \cup \gtrdot$).

Proposition 3.15 (cf. [21, Theorem 3.3])**.** *Let $\beta \in \mathrm{FO}(\mathrm{N}, \overrightarrow{Z}, \lessdot)$. There exist integers $\ell \geq 0$, $r \geq 1$ and a formula $\gamma(x_1, \ldots, x_\ell, y) \in \mathrm{FO}(\mathrm{N}, \overrightarrow{Z}, \lessdot)$ that is r-local around y such that for any structure $t = (V, \lessdot, (P_a)_{a\in N}, \overrightarrow{Z})$, we have*

$$t \models \beta \quad \textit{iff} \quad t \models \exists x_1 \cdots \exists x_\ell \forall y\, \gamma.$$

Note that these two formulas are in particular equivalent for any trace t whatever the dependence alphabet is.

Remark 3.16. Keisler and Lotfallah [13, Corollary 6.2] gave bounds for ℓ and r in the above proposition: Let r be the minimal integer of the form $n \cdot 4^n$ such that the quantifier rank of β is at most $\log(r) + 1$. Then β is equivalent to a finite conjunction of formulas $\exists x_1 \cdots \exists x_\ell \forall y\, \gamma$ with $\ell \leqslant n$ and γ r-local around y. This finite conjunction can then be brought into the above form at the expense of a larger value of ℓ.

We will build an automaton for the formula $\forall y\, \gamma$ in space $\mathtt{poly}(|\Sigma|)$. The idea is, reading a word w, to compute the top $2r$-spheres of all its prefixes with an automaton. Then, using the results in the previous section, we are able to check all r-spheres in the trace $[w]$. Since γ is r-local around y, its truth value depends on the r-sphere around y and also on the truth values of atomic propositions involving only variables that are free in $\forall y\, \gamma$, such as $x_i \lessdot x_j$ or $x_i = x_j$ or $x_i \in Z_j$ or $\lambda(x_i) = a$. We will guess the truth values of these atomic propositions so that we can check $\forall y\, \gamma$ just knowing the r-spheres. Let H be the set of atomic propositions in γ involving only variables that are free in $\forall y\, \gamma$. Then the formula $\forall y\, \gamma$ is equivalent to a disjunction $\bigvee_{E\subseteq H} \gamma_E^1 \wedge \forall y\, \gamma_E^2$ where $\gamma_E^1 = \bigwedge_{\delta\in E} \delta \wedge \bigwedge_{\delta\in H\setminus E} \neg\delta$ and γ_E^2 is obtained from γ by replacing any occurrence of $\delta \in E$ with true and $\delta \in H\setminus E$ with false. Note that H does not depend on Σ so the number of elements in the disjunction $\bigvee_{E\subseteq H} \gamma_E^1 \wedge \forall y\, \gamma_E^2$ is constant.

We fix some $E \subseteq H$ and define the automaton $\mathcal{A}_E$ for the formula $\gamma_E^1 \wedge \forall y\, \gamma_E^2$. The free variables in this formula are $\overrightarrow{Z} = (Z_1, \ldots, Z_p)$ and $\overrightarrow{x} = (x_1, \ldots, x_\ell)$ hence the automaton needs to read words over the alphabet $\Sigma' = \Sigma \times \{0, 1\}^{p+\ell}$. As in the beginning of this section, we will write $\overline{w} = (w, \overrightarrow{Z}, \overrightarrow{x})$ a word over Σ'. Actually, the lines for the variables $x_1, \ldots, x_\ell$ define sets. The automaton $\mathcal{A}_E$ will check that these sets are singletons so that they define the assignment of the first order variables as usual.

[3] Actually, [8, Proposition 7.3] gives a more precise space bound using the notion of *special variance*. Here, we only use the fact that the special variance is always bounded by the number of states of the automaton.

[4] Schwentick and Bartelmann [21, Theorem 3.3] presuppose a finite signature. But one can check that what is really needed is that there are only finitely many non-unary predicates. Since $\lessdot$ is the only such relation in our signature, the result can indeed be applied here.

A *state* of $\mathcal{A}_E$ is a tuple $q = (\mathcal{M}, (V_i)_{1\leqslant i\leqslant p+\ell}, B, C, (\varepsilon_i)_{1\leqslant i\leqslant \ell})$ satisfying the following conditions:

(S1) $\mathcal{M} = \mathrm{TOP}_{2r}(s) = (W, \leqslant, \lessdot, \lambda)$ for some finite trace $s = (V, \leq, \lambda) \in \mathbb{M}(\Sigma, D)$ (the intuition is that $\mathcal{M} = \mathrm{TOP}_{2r}([w])$ if the automaton has read a finite word $\overline{w} = (w, \overrightarrow{Z}, \overrightarrow{x})$),
(S2) $V_i \subseteq W$ for each $1 \leqslant i \leqslant p+\ell$ (the intuition is that V_i is the intersection of W with the set defined by the i-th line of $\overline{w}$), and $|V_{p+i}| \leqslant 1$ for $1 \leqslant i \leq \ell$,
(S3) $B, C \subseteq \Sigma$ (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),
(S4) $\varepsilon_i \in \{0, 1\}$ is a flag signaling whether a one has already been seen on the line for the variable x_i; it will be used to check that these lines define singletons.

A state $q = (\mathcal{M}, (V_i)_{1\leqslant i\leqslant p+\ell}, B, C, (\varepsilon_i)_{1\leqslant i\leqslant \ell})$ is *initial* if $\mathcal{M} = \mathrm{TOP}_{2r}(\varepsilon)$ is empty (which implies that each $V_i = \emptyset$), and C is empty and $\varepsilon_i = 0$ for each $1 \leqslant i \leqslant \ell$. The state q is *accepting* if $C = \emptyset$ and $\varepsilon_i = 1$ for each $1 \leqslant i \leqslant \ell$.

Before defining the transitions of $\mathcal{A}_E$, we give two definitions keeping the notations as above, in particular, $q = (\mathrm{TOP}_{2r}(s), (V_i)_{1\leqslant i\leqslant p+\ell}, B, C, (\varepsilon_i)_{1\leqslant i\leqslant \ell})$ is a state and $a \in \Sigma$ is a letter. We say that a vertex $v \in W$ is r-safe for q if it is r-safe for (s, B) (recall that being r-safe for (s, B) only depends on B and $\mathrm{TOP}_r(s)$ which can be determined from q). Next, we say that a vertex $v \in W$ is r-critical for (q, a) if it is r-critical for (s, a) (recall that being r-critical for (s, a) only depends on a and $\mathrm{TOP}_r(s)$ which can be determined from q).

To define the *transitions* of $\mathcal{A}_E$, let $\overline{a} = (a, (b_i)_{1\leqslant i\leqslant p+\ell}) \in \Sigma'$ be a letter and consider two states $q = (\mathcal{M}, (V_i)_{1\leqslant i\leqslant p+\ell}, B, C, (\varepsilon_i)_{1\leqslant i\leqslant \ell})$ and $q' = (\mathcal{M}', (V'_i)_{1\leqslant i\leqslant p+\ell}, B', C', (\varepsilon'_i)_{1\leqslant i\leqslant \ell})$ of the automaton $\mathcal{A}_E$. There is a transition $q \xrightarrow{\overline{a}} q'$ in $\mathcal{A}_E$ iff the following conditions hold:

(T1) $\mathcal{M} = (W, \leqslant, \lessdot, \lambda) = \mathrm{TOP}_{2r}(s)$ and $\mathcal{M}' = (W', \leqslant, \lessdot, \lambda) = \mathrm{TOP}_{2r}(sa)$ for some finite trace $s \in \mathbb{M}(\Sigma, D)$ (by Lemma 3.7, $\mathcal{M}'$ is uniquely defined by $\mathcal{M}$ and the letter a),
(T2) $V'_i = V_i \cap W'$ if $b_i = 0$ and $V'_i = (V_i \cap W') \uplus (W' \backslash W)$ if $b_i = 1$ (note that, by the proof of Lemma 3.7, $W' \backslash W$ is a singleton corresponding to the added letter a),
(T3) $B = B' \cup \{a\}$ (thus, B' can be chosen non-deterministically),
(T4) $C' = C \backslash \{a\}$ if $C \neq \emptyset$ and $C' = B'$ otherwise,
(T5) $\varepsilon'_i = \varepsilon_i + b_{p+i}$ (in particular, there is no transition $q \xrightarrow{\overline{a}} q'$ if $\varepsilon_i = b_{p+i} = 1$),
(T6) If $v \in W$ is r-safe for q or r-critical for (q, a) and $V_{p+i} = \{v\}$ for some $1 \leqslant i \leqslant \ell$, then for each $\delta \in H$ in which x_i occurs, we have $\delta \in E$ if and only if one of the following hold:
- $\delta = (\lambda(x_i) = b)$ and $\lambda(v) = b$,
- $\delta = (x_i \in Z_n)$ and $v \in V_n$,
- $\delta = (x_i = x_n)$ and $V_{p+n} = \{v\}$,
- $\delta = (x_i \lessdot x_n)$, $V_{p+n} = \{v_n\}$ is a singleton, and $v \lessdot v_n$ in $\mathcal{M}$.

Lemma 3.11 ensures that the r-sphere around v is contained in $\mathcal{M}$, hence legitimates the last constraint.
(T7) If $v \in W$ is r-safe for q or r-critical for (q, a), then $\mathcal{M}, V_1, \ldots, V_{p+\ell}, v \models \gamma_E^2$. Here, $V_1, \ldots, V_p$ and v are the assignments for the free variables $Z_1, \ldots, Z_p$ and y. We have to explain how to evaluate $\mathcal{M}, V_1, \ldots, V_{p+\ell}, v \models \gamma_E^2$ although some sets V_{p+i} may be empty and do not define a proper assignment for the first order variable x_i. Note that if $V_{p+i} \neq \emptyset$ then it is a singleton $\{v_i\}$ which encodes the assignment for x_i. Hence, the only difficulty is when $V_{p+i} = \emptyset$. In this case, we evaluate to false all atomic propositions of γ_E^2 in which x_i occurs. Note that such atomic propositions must be of the form $x_i = z$ or $x_i \lessdot z$ or $z \lessdot x_i$ where z is either y or a variable that is bound in γ_E^2. Also, since $V_{p+i} = \emptyset$, the assignment of x_i is not in the r-sphere of v and since γ_E^2 is r-local around y, the assignment u for z satisfies $d_{\bowtie}(u, v) < r$. Hence, the evaluation of these atomic propositions to false is justified.

Let $\mathcal{A}_E$ denote the Büchi-automaton $(Q, \Sigma', I, F, \to)$ defined so far. Since the essential information in a state is the first component, i.e., a sphere in a trace, we will speak of the *sphere automaton*. The only non-determinism in the automaton $\mathcal{A}_E$ comes from the component B of the state but in fact, the automaton is unambiguous.

Proposition 3.17. *Let $\overline{w} = (w, (Z_i)_{1\leqslant i\leqslant p+\ell}) \in \Sigma'^{\omega}$. Then $\overline{w}$ is accepted by $\mathcal{A}_E$ if and only if each $Z_{p+i} = \{x_i\}$ is a singleton set for $1 \leqslant i \leqslant \ell$ and*

$$[w], Z_1, \ldots, Z_p, x_1, \ldots, x_\ell \models \gamma_E^1 \wedge \forall y\, \gamma_E^2.$$

Proof. Assume first that $\overline{w}$ is accepted by $\mathcal{A}_E$. Write $w = a_1 a_2 \cdots$ and let $[w] = (V, \leqslant, \lambda) \in \mathbb{R}(\Sigma, D)$. Consider an accepting run $q^0 \xrightarrow{\overline{a}_1} q^1 \xrightarrow{\overline{a}_2} q^2 \cdots$ for $\overline{w}$ in $\mathcal{A}_E$. Write $q^k = (\mathcal{M}^k, (V_i^k)_{1\leqslant i\leqslant p+\ell}, B^k, C^k, (\varepsilon_i^k)_{1\leqslant i\leqslant \ell})$. By definition of the transition function and the initial states we have $\mathcal{M}^k = \mathrm{TOP}_{2r}([a_1 \cdots a_k]) = (W^k, \leqslant, \lessdot, \lambda)$ and $V_i^k = Z_i \cap W^k$. Since the run is accepting, the set $Z_{p+i} = \{x_i\}$ is a singleton for $1 \leqslant i \leqslant \ell$. Moreover, $B^k = \mathrm{alph}(a_{k+1} a_{k+2} \cdots)$ for all $k \geqslant 0$: Clearly,

B_k must contain all letters that remain to be read. Conversely, if for some $k \geqslant 0$,the set B_k contains some additional letters then we can check that $C_j = \emptyset$ for at most one $j \geqslant k$, a contradiction. Therefore, $\mathcal{A}_E$ is unambiguous.

For each vertex v of $[w]$, we apply Lemma 3.12 and we find $k \geqslant 1$ such that v is either r-critical for $([a_1 \cdots a_k], a_{k+1})$ or r-safe for $([a_1 \cdots a_k], \mathrm{alph}(a_{k+1}a_{k+2}\ldots))$. From (T7), we get $\mathcal{M}^k, V_1^k, \ldots, V_{p+\ell}^k, v \models \gamma_E^2$ (recall that atomic propositions of γ_E^2 in which x_i occurs are evaluated to false if $V_{p+i}^k = \emptyset$). We show that this implies $[w], \overrightarrow{Z}, \overrightarrow{x}, v \models \gamma_E^2$. Indeed, the formula γ_E^2 is r-local around y and by Lemmas 3.9 and 3.11 we know that $\mathrm{SPH}_r([w], v) = \mathrm{SPH}_r(\mathrm{TOP}_{2r}([a_1 \cdots a_k]), v) = \mathrm{SPH}_r(\mathcal{M}^k, v)$. Moreover, we have seen that $V_i^k = Z_i \cap W^k$. Finally, let $1 \leqslant i \leqslant \ell$ and assume that $V_{p+i}^k = \emptyset$. An atomic proposition of γ_E^2 in which x_i occurs must be of the form $x_i = z$ or $x_i \lessdot z$ or $z \lessdot x_i$ where z is either y or a variable that is bound in γ_E^2. Since $V_{p+i} = \emptyset$, we have $d_{G([w])}(v, x_i) > r$ and because γ_E^2 is r-local around y we know that the assignment u of z satisfies $d_{G([w])}(u, v) < r$. Hence, these atomic propositions evaluate to false in the context $[w], \overrightarrow{Z}, \overrightarrow{x}, v$. We deduce that $[w], \overrightarrow{Z}, \overrightarrow{x}, v \models \gamma_E^2$. Since this holds for each vertex v of $[w]$ we obtain $[w], \overrightarrow{Z}, \overrightarrow{x} \models \forall y\, \gamma_E^2$.

For each $1 \leqslant i \leqslant \ell$, we apply Lemma 3.12 to the vertex x_i of $[w]$ and we find $k \geqslant 1$ such that x_i is either r-critical for $([a_1 \cdots a_k], a_{k+1})$ or r-safe for $([a_1 \cdots a_k], \mathrm{alph}(a_{k+1}a_{k+2}\ldots))$. We have $\mathrm{SPH}_r([w], x_i) = \mathrm{SPH}_r(\mathcal{M}^k, x_i)$ in either case and in particular $V_{p+i}^k = \{x_i\} \cap W^k = \{x_i\}$. Therefore, by (T6), each conjunct of γ_E^1 must evaluate to true. We deduce that $[w], \overrightarrow{Z}, \overrightarrow{x} \models \gamma_E^1$.

Conversely, assume that $Z_{p+i} = \{x_i\}$ is a singleton set for $1 \leqslant i \leqslant \ell$ and that $[w], \overrightarrow{Z}, \overrightarrow{x} \models \gamma_E^1 \wedge \forall y\, \gamma_E^2$. Write $w = a_1a_2 \cdots$ and let $[w] = (V, \leqslant, \lambda) \in \mathbb{R}(\Sigma, D)$. We show that $\overline{w} = (w, (Z_i)_{1 \leqslant i \leqslant p+\ell})$ is accepted by $\mathcal{A}_E$. We consider temporarily the automaton $\mathcal{A}'_E$ defined as $\mathcal{A}_E$ without (T6, T7).

Let q^0 be the unique initial state with B-component $\mathrm{alph}(w)$. By definition of the transition function, there is exactly one run $q^0 \xrightarrow{\overline{a}_1} q^1 \xrightarrow{\overline{a}_2} q^2 \cdots$ for $\overline{w}$ in $\mathcal{A}'_E$ such that the B-component of q^k is $\mathrm{alph}(a_{k+1}a_{k+2}\cdots)$. As above, we write $q^k = (\mathcal{M}^k, (V_i^k)_{1 \leqslant i \leqslant p+\ell}, B^k, C^k, (\varepsilon_i^k)_{1 \leqslant i \leqslant \ell})$. By definition of the transition function and the initial states we have $\mathcal{M}^k = \mathrm{TOP}_{2r}([a_1 \cdots a_k]) = (W^k, \leqslant, \lessdot, \lambda)$ and $V_i^k = Z_i \cap W^k$. Also, since the sets Z_{p+i} are singletons, there is some K such that for each $k > K$ and $1 \leqslant i \leqslant \ell$ we have $\varepsilon_i^k = 1$. Finally, since $B^k = \mathrm{alph}(a_{k+1}a_{k+2}\cdots)$ we deduce from the definition of the transition function that $C^k = \emptyset$ for infinitely many k's. Therefore, the run is accepting in $\mathcal{A}'_E$.

It remains to show that this run is actually in $\mathcal{A}_E$, i.e., that all transitions satisfy (T6, T7).

(T6) Let $1 \leqslant i \leqslant \ell$ and assume that $V_{p+i} = \{v_i\}$ is a singleton and that v_i is either r-critical for (q_k, a_{k+1}) or r-safe for q_k. We must have $v_i = x_i$. By Lemmas 3.9 and 3.11 we know that $\mathrm{SPH}_r([w], x_i) = \mathrm{SPH}_r(\mathcal{M}^k, x_i)$. Since $[w], \overrightarrow{Z}, \overrightarrow{x} \models \gamma_E^1$, we deduce that (T6) is fulfilled.

(T7) Assume that $v \in W^k$ is either r-critical for (q_k, a_{k+1}) or r-safe for q_k. By hypothesis, we have $[w], \overrightarrow{Z}, \overrightarrow{x}, v \models \gamma_E^2$. We can show as in the first part of the proof that this implies $\mathcal{M}^k, V_1^k, \ldots, V_{p+\ell}^k, v \models \gamma_E^2$ (with atomic propositions of γ_E^2 in which x_i occurs evaluated to false if $V_{p+i}^k = \emptyset$). Therefore, (T7) is fulfilled.

We deduce that $\overline{w}$ is accepted by $\mathcal{A}_E$ as required. □

Lemma 3.18. *Given Σ, the sphere automaton $\mathcal{A}_E$ can be constructed in space $\mathtt{poly}(|\Sigma|)$. Hence, the number of states of $\mathcal{A}_E$ is in $2^{\mathtt{poly}(|\Sigma|)}$.*

Proof. Let $s = (V, \leqslant, \lambda) \in \mathbb{M}(\Sigma, D)$. Any node $v \in V$ has at most h many $\lessdot$-successors and h many $\lessdot$-predecessors, where h is the size of the largest independence clique that is contained in some $D(a)$ for $a \in \Sigma$. Clearly, $h \leq |\Sigma|$ though it is usually much smaller, e.g., $h = 1$ for words. Thus, the number of nodes at distance k from v is at most $(2h)^k$. Hence, the number of nodes in $\mathrm{TOP}_{2r}(s)$ is at most $K = |\Sigma| \left(1 + 2h + \cdots + (2h)^{2r}\right)$. Since $h \leqslant |\Sigma|$, we get $K \leqslant (2|\Sigma|)^{2r+1}$. We deduce that the Σ-labeled graph $\mathrm{TOP}_{2r}(s)$ with $n \leqslant K$ nodes and two edge relations $\lessdot$ and $\leq$ can be stored in space $\mathtt{poly}(|\Sigma|)$. Therefore, a state $q = (\mathcal{M}, (V_i)_{1 \leqslant i \leqslant p+\ell}, B, C, (\varepsilon_i)_{1 \leqslant i \leqslant \ell})$ of the sphere automaton can also be stored in space $\mathtt{poly}(|\Sigma|)$ (note that p and ℓ are constants which do not depend on Σ).

Now, from its definition, we can easily check that the transition relation of $\mathcal{A}_E$ can be decided in space $\mathtt{poly}(|\Sigma|)$, i.e., given states $q, q' \in Q$ and a letter $\overline{a} \in \Sigma'$, we can check whether there is a transition $q \xrightarrow{\overline{a}} q'$ in $\mathcal{A}_E$ in space $\mathtt{poly}(|\Sigma|)$. Therefore, we can enumerate all transitions of $\mathcal{A}_E$ in space $\mathtt{poly}(|\Sigma|)$ (simply enumerate the triples $(q, \overline{a}, q')$ and for each of them check whether it is a valid transition of $\mathcal{A}_E$).

We deduce that, given Σ, the sphere automaton $\mathcal{A}_E$ can be constructed in space $\mathtt{poly}(|\Sigma|)$. □

Proof of Proposition 3.13. Recall that the formula $\forall y\, \gamma$ is equivalent to a disjunction $\bigvee_{E \subseteq H} \gamma_E^1 \wedge \forall y\, \gamma_E^2$ and the number of elements in this disjunction does not depend on Σ. From Proposition 3.17 we deduce that we can construct an automaton $\mathcal{A}_\gamma$ over the alphabet Σ' for the formula $\forall y\, \gamma$ as a disjoint union of the sphere automata $\mathcal{A}_E$. Using Lemma 3.18 we know

that $\mathcal{A}_\gamma$ can be constructed in space $\mathtt{poly}(|\Sigma|)$. Projecting $\mathcal{A}_\gamma$ to the subalphabet $\Sigma \times \{0,1\}^p$ of Σ' we obtain an automaton $\mathcal{B}$ for the formula β which is equivalent to $\exists x_1 \cdots \exists x_\ell \forall y\ \gamma$. Again, $\mathcal{B}$ can be constructed in space $\mathtt{poly}(|\Sigma|)$.

Recall that $\varphi(X_1, \ldots, X_m)$ is equivalent to

$$\exists^{(\mathrm{fin})} \overrightarrow{Y_1} \neg\exists^{(\mathrm{fin})} \overrightarrow{Y_2} \neg\exists^{(\mathrm{fin})} \overrightarrow{Y_3} \cdots \neg\exists^{(\mathrm{fin})} \overrightarrow{Y_n}\ \beta\left(X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n}\right).$$

Using the following classical constructions on automata, we can construct the automaton $\mathcal{A}_\varphi$:

- projection for existential quantification $\exists$,
- intersection with $(\Sigma \times \{0,1\}^j)^*(\Sigma \times \{0,1\}^{j-1} \times \{0\})^\omega$ and projection for existential finite-set quantification $\exists^{\mathrm{fin}}$,
- complement for negation.

Note that each complement needs an exponential: as in the proof of Proposition 3.4, if a Büchi automaton $\mathcal{B}$ can be constructed in space $\mathtt{tower}(k, \mathtt{poly}(|\Sigma|))$ then it has at most $\mathtt{tower}(k+1, \mathtt{poly}(|\Sigma|))$ many states and by Proposition 3.19 we can construct a Büchi automaton for the complement of $\mathcal{L}(\mathcal{B})$ in space $\mathtt{poly}(\mathtt{tower}(k+1, \mathtt{poly}(|\Sigma|))) = \mathtt{tower}(k+1, \mathtt{poly}(|\Sigma|))$. Therefore, the automaton $\mathcal{A}_\varphi$ can be constructed in space $\mathtt{tower}(n-1, \mathtt{poly}(|\Sigma|))$. □

3.4. Construction of modality automata

Now we have almost all ingredients for the proof of Proposition 3.4. The only one that is still missing is the effective complementation of Büchi-automata from [15]. We also sketch its proof in order to state precisely its complexity.

Proposition 3.19. *Let $\mathcal{B} = (Q, \iota, T, F)$ be a Büchi-automaton over the alphabet Σ. Then, in space $\mathtt{poly}(|Q|)$, one can compute a Büchi-automaton $\mathcal{C}$ over Σ such that $\mathcal{L}(\mathcal{C}) = \Sigma^\omega \setminus \mathcal{L}(\mathcal{B})$.*

Proof. To obtain the automaton $\mathcal{C}$, we consider $\mathcal{B}$ as an alternating Büchi-automaton, i.e., $\mathcal{B} = (Q, \iota, \delta, F)$ with

$$\delta(p,a) = \bigvee_{(p,a,q)\in T} q$$

for all $p \in Q$ and $a \in \Sigma$. From this alternating Büchi-automaton, we obtain an alternating co-Büchi-automaton $\mathcal{B}_1 = (Q, \iota, \delta_1, F)$ with $\mathcal{L}(\mathcal{B}_1) = \Sigma^\omega \setminus \mathcal{L}(\mathcal{B})$ setting

$$\delta_1(p,a) = \bigwedge_{(p,a,q)\in T} q$$

for all $p \in Q$ and $a \in \Sigma$. Then, by [15], this alternating co-Büchi automaton can be transformed into an equivalent weak alternating automaton $\mathcal{B}_2$ whose set of states equals $Q \times \{0, 1, \ldots, 2|Q|\}$. The transitions of this automaton are given by

$$\delta_2((p,n),a) = \begin{cases} \bigwedge_{(p,a,q)\in T} \bigvee_{n' \leqslant n} (q, n') & \text{if } p \in F \text{ or } n \text{ is even} \\ \text{false} & \text{otherwise.} \end{cases}$$

Adapting the proof from [18], one can construct an equivalent Büchi-automaton $\mathcal{C}$ whose states consist of two subsets of $Q' = Q \times \{0, 1, \ldots, 2|Q|\}$. To store one such state, space $2|Q'| \in \mathtt{poly}(|Q|)$ suffices. Moreover, from the construction of [18], one can see that the transition function of $\mathcal{C}$ can be decided in space $O(|Q'|) = O(|Q|)$ from $\mathcal{B}$. Finally, $\mathcal{L}(\mathcal{C}) = \mathcal{L}(\mathcal{B}_2) = \mathcal{L}(\mathcal{B}_1) = \Sigma^\omega \setminus \mathcal{L}(\mathcal{B})$. □

We are now ready to close the main gap in the proof of Theorem 3.1.

Proof of Proposition 3.4. First, the formula

$$\alpha'(X_0, X_1, \ldots, X_m) = \forall x\ (x \in X_0 \leftrightarrow \alpha(X_1, \ldots, X_m, x))$$

can be equivalently written as

$$\forall x\ (x \notin X_0 \vee \alpha(X_1, \ldots, X_m, x)) \wedge \neg\exists x\ (\alpha(X_1, \ldots, X_m, x) \wedge x \notin X_0).$$

Since $\alpha \in \mathrm{M}\Sigma^1_n(\mathrm{N}, <)$, this is a conjunction of a formula of the form $\forall x\, \varphi$ and a formula of the form $\neg\psi$ with $\varphi, \psi \in \mathrm{M}\Sigma^1_n(\mathrm{N}, <)$. From Proposition 3.14 (page 805), we can construct a Büchi automaton $\mathcal{B}_\varphi$ for the first conjunct $\forall x\, \varphi$ in space $\mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$. From Proposition 3.13 (page 805), we can construct a Büchi automaton $\mathcal{B}$ for ψ in space $\mathtt{tower}(n-$

$1, \mathtt{poly}(|\Sigma|))$. We deduce that the number of states of $\mathcal{B}$ is in $\mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$. Using Propositions 3.19 we can construct a Büchi automaton for the second conjunct $\neg\psi$ in space $\mathtt{poly}(\mathtt{tower}(n, \mathtt{poly}(|\Sigma|))) = \mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$. The final construction for the intersection does not change the space bound. □

4. n-EXPSPACE lower bound for $\mathrm{M}\Pi^1_n(\mathrm{N}, \lessdot)$-logics

This section is devoted to the proof of the following.

Theorem 4.1. *Let $n \geqslant 1$. There is an $\mathrm{M}\Pi^1_n(\mathrm{N}, \lessdot)$-definable temporal logic TL_n such that its uniform satisfiability problem is* **n-EXPSPACE***-hard (and therefore* **n-EXPSPACE***-complete by Theorem 3.1).*

Towards this aim, we will restrict ourselves to finite traces. Consider the $\mathrm{M}\Pi^1_1(\mathrm{N}, \lessdot)$-formula

$$[\![\text{finite}]\!] = \forall X(X = \emptyset \vee \exists x(x \in X \wedge \forall y(x \lessdot y \to y \notin X))).$$

Since any infinite trace t over a finite dependence alphabet admits an infinite path $x_0 \lessdot x_1 \lessdot x_2 \ldots$, this formula holds in t iff t is finite. Adding it as a constant modality to some $\mathrm{M}\Pi^1_n(\mathrm{N}, \lessdot)$-logic $\mathrm{TL}(B)$ reduces the *finite* uniform satisfiability problem of $\mathrm{TL}(B)$ to the uniform satisfiability problem of the extended temporal logic $\mathrm{TL}(B \cup \{\text{finite}\})$. Thus, restricting attention to finite traces is at least as complicated as the general case.

We consider functions $F_n : \mathbb{N} \to \mathbb{N}$ that are defined inductively by $F_0(m) = m$ and $F_{n+1}(m) = F_n(m) \cdot 2^{F_n(m)}$ for $n \geq 0$. For $m \geqslant 1$ and $n \geqslant 0$ we have $\mathtt{tower}(n, m) \leqslant F_n(m)$. Hence, there is a Turing machine M that runs in space $F_n(m) - 3$ (where m is the input-size) and accepts some **n-EXPSPACE**-hard problem. Then, Theorem 4.1 can be proved by a polynomial reduction of the language of this Turing machine to the satisfiability problem of some temporal logic TL_n to be defined later.

Notation. Let Γ_{tape} be the tape alphabet including the blank symbol $\square$ and the end-of-tape markers $\triangleright$ and $\triangleleft$ and let Q be the set of states of the Turing machine M. We will write $\Gamma = \Gamma_{\text{tape}} \uplus Q$ for the alphabet of the Turing machine M. For $m \geqslant 1$ (m will be the length of the input word), an m-configuration is a word $\triangleright\alpha q\beta\triangleleft$ of length $F_n(m)$ where $\alpha\beta \in (\Gamma_{\text{tape}} \setminus \{\triangleright, \triangleleft\})^*$ is the tape content and q is the state of the Turing machine. The intuition is that the head is on the first letter of $\beta\triangleleft$. An m-computation is a word $c_0\, c_1\, c_2 \ldots c_k$ where c_i are m-configurations with $c_i \vdash_M c_{i+1}$ for all $0 \leqslant i < k$. Note that there is a relation $R \subseteq \Gamma^6$ such that a word

$$w \in \left(\Gamma^{F_n(m)} \cap (\triangleright(\Gamma_{\text{tape}} \setminus \{\triangleright, \triangleleft\})^* Q (\Gamma_{\text{tape}} \setminus \{\triangleright, \triangleleft\})^* \triangleleft)\right)^+$$

is an m-computation if and only if it satisfies

$$w \in \Gamma^* \gamma_1\gamma_2\gamma_3 \Gamma^{F_n(m)-3} \delta_1\delta_2\delta_3 \Gamma^* \Rightarrow (\gamma_1, \gamma_2, \gamma_3, \delta_1, \delta_2, \delta_3) \in R$$

for all $\gamma_1, \gamma_2, \gamma_3, \delta_1, \delta_2, \delta_3 \in \Gamma$.

We will encode these computations by interspersing them with letters from another alphabet. So let A be some countably infinite alphabet with $A \cap \Gamma = \emptyset$. As abbreviation, we use the infinite alphabet $\Sigma = \Gamma \cup A$ and denote by π_Γ the projection from Σ^* to Γ^*. Then, for $m \geqslant 1$, we define the language

$$L_m = \bigcup (a_1 \Gamma a_2 \Gamma \cdots a_m \Gamma)^* a_1 \Gamma a_2 \Gamma \cdots \Gamma a_k$$

where the union ranges over all $a_1, \ldots, a_m \in A$ which are pairwise distinct and all $1 \leqslant k \leqslant m$. We also define $L = \bigcup_{m \geq 1} L_m$ and the set

$$C = \{w \in L \mid \pi_\Gamma(w) \text{ is an } m\text{-computation for some } m \geqslant 1\}$$

which serves as encoding of the set of computations of M. Section 4.1 deals with this set of words, Section 4.2 will give a further encoding into 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_n (that we are going to construct from the Turing machine M) and a finite 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

In this section, we will consider formulas whose models are words over the alphabet Σ. The syntax of our monadic second order logic $\mathrm{MSO}(\Gamma, \lessdot, \prec)$ is given by

$$\varphi ::= x \in X \mid \lambda(x) = \gamma \mid x \lessdot y \mid x \prec y \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x\varphi \mid \exists X\varphi$$

where x and y are individual variables, X is a set variable, and $\gamma \in \Gamma$ is a letter from Γ. Individual variables range over positions in a word and set variables over sets of positions in a word. Note that formulas $\lambda(x) = a$ for $a \in A$ are not allowed. More formally, the set of positions of a finite word $w \in \Sigma^*$ is $\mathrm{pos}(w) = \{i \mid 0 \leqslant i < |w|\}$. Let $w = a_0 a_1 \dots a_{|w|-1}$ with $a_i \in \Sigma$ and $x, y \in \mathrm{pos}(w)$. Then we define

$$\begin{array}{ll} w \models \lambda(x) = \gamma & \text{if} \quad a_x = \gamma \\ w \models x \lessdot y & \text{if} \quad y = x + 1 \\ w \models x \prec y & \text{if} \quad a_x = a_y \in A \text{ and } a_z \neq a_x \text{ for all } x < z < y \end{array}$$

Note that $x \prec y$ cannot be expressed in $\mathrm{FO}(\Gamma, <)$ or in $\mathrm{MSO}(\Gamma, \lessdot)$ since we cannot express $\lambda(x) = \lambda(y)$ when $\lambda(x) \in A$. We will freely use formulas like $\lambda(x) \in E$ for $E \subseteq \Gamma$ meaning $\bigvee_{e \in E} \lambda(x) = e$.

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

Proof. Let α be the following formula:

$$\begin{aligned} & \exists x, y\, (\lambda(x) \notin \Gamma \wedge \lambda(y) \notin \Gamma \wedge \neg \exists z\, (z \lessdot x \vee y \lessdot z)) \\ \wedge\ & \forall x, y\, (x \lessdot y \to (\lambda(x) \in \Gamma \leftrightarrow \lambda(y) \notin \Gamma) \\ \wedge\ & \forall x, y, x', y', x'', y''\, \Big(x \lessdot x' \lessdot x'' \wedge y \lessdot y' \lessdot y'' \to \big(x \prec y \leftrightarrow x'' \prec y''\big)\Big) \end{aligned}$$

The first line of the formula expresses that the first and last letters of a word from Σ^* belong to A, i.e., it defines the language $A\Sigma^*A \cup A$. The second line expresses that letters from Γ and A alternate, together with the first line, it defines the language $(A\Gamma)^*A$. So let $w = a_1\gamma_1 a_2\gamma_2 \dots a_{k-1}\gamma_{k-1}a_k$ be a word from this set with $a_i \in A$ and $\gamma_i \in \Gamma$. Then the premise in the last line expresses $x'' = x + 2$ and $y'' = y + 2$. Hence, the last formula is satisfied by w iff the projection of w to A^* is the prefix of some word v^ℓ where no letter in v occurs twice. In summary, this $\mathrm{FO}(\Gamma, \lessdot, \prec)$-formula is satisfied by w iff $w \in L$. □

Lemma 4.3. *There is a formula* $\mathrm{interval}(x, y, X)$ *in* $\mathrm{FO}(\Gamma, \lessdot, \prec)$ *such that, for any finite word $w \in \Sigma^*$, any $x, y \in \mathrm{pos}(w)$ and $X \subseteq \mathrm{pos}(w)$, we have $w \models \mathrm{interval}(x, y, X)$ iff $x \leqslant y$ and $X = \{x, x + 1, \dots, y\}$.*

Proof. Let $\mathrm{interval}(x, y, X)$ denotes the following formula:

$$\begin{aligned} & x \in X \wedge \forall x'(x' \in X \to x' = x \vee \exists y'(y' \lessdot x' \wedge y' \in X)) \\ \wedge\ & y \in X \wedge \forall x'(x' \in X \to x' = y \vee \exists y'(x' \lessdot y' \wedge y' \in X)). \end{aligned}$$

Suppose $w \models \mathrm{interval}(x, y, X)$. The first line expresses that the set X is a nonempty downwards closed subset of $\{x, x + 1, \dots, |w| - 1\}$ while the second line expresses that X is a nonempty upwards closed subset of $\{0, 1, \dots, y\}$. In other words, $X = \{x, x + 1, \dots, y\}$ as required. □

Lemma 4.4. *For all $n \geqslant 0$, there exists a formula $\varphi_n(x, y) \in \mathrm{M}\Sigma_n^1(\Gamma, \lessdot, \prec)$ with two free individual variables x and y such that, for all $m \geqslant 1$, $w \in L_m$ and $k, \ell \in \mathrm{pos}(w)$, we have $w \models \varphi_n(k, \ell)$ iff k is even and $\ell = k + 2F_n(m)$.*

The idea of the inductive proof is to split the interval $[k, \ell)$ into blocks of length $2F_{n-1}(m)$ and to encode, in these blocks, the binary representations of the numbers $0, 1, \dots, 2^{F_{n-1}(m)} - 1$. This idea was first used by Matz [16] (cf. also [17]) for pictures and is significantly different from Walukiewicz's method of nested counters [24].[5]

Proof. For $n = 0$, we set

$$\varphi_0(x, y) = (x \prec y).$$

Let $w = a_0 a_1 \dots a_{|w|-1} \in L_m$ and $k, \ell \in \mathrm{pos}(w)$. Then $w \models \varphi_0(k, \ell)$ iff $k \prec \ell$, i.e., iff $a_k = a_\ell \in A$, $k < \ell$, and there is no occurrence of a_k in between. Since $w \in L_m$, this is equivalent to saying k is even and $\ell = k + 2m$, i.e., $\ell = k + 2F_0(m)$.

For $n \geq 0$, let $\varphi_{n+1}(x, y)$ denotes the following formula (we advise to read the explanations below and look at Figs. 3 and 4 simultaneously with each line of the formula):

[5] Using Walukiewicz's method in a previous version of this paper, we needed one more quantifier alternation resulting in an exponentially weaker lower bound in Theorem 4.1.

Fig. 3. Conjuncts (1–8). This picture visualizes some word with positions x, x', y', and y. There are sets X_0 and Y_0 that contain all the positions between x and x' and y' and y, respectively, The nodes drawn (including x, x', etc.) form the set Z. Furthermore, there is a set B ("B" stands for "bit"): it contains all positions marked 1, none marked 0, and possibly some positions in between. Finally, the top line indicates that successive elements of Z (i.e., drawn positions) have distance $2F_n(m)$. Conjuncts (1–8) ensure this situation (as well as the fact that X contains all the positions between x and y).

$$\exists X, X_0, Y_0, Z, B, x', y' :$$
$$\text{interval}(x, y, X) \tag{1}$$
$$\wedge\ \text{interval}(x, x', X_0) \wedge \varphi_n(x, x') \wedge X_0 \subseteq X \tag{2}$$
$$\wedge\ \text{interval}(y', y, Y_0) \wedge \varphi_n(y', y) \tag{3}$$
$$\wedge\ Z \cap X_0 = \{x, x'\} \wedge \forall z, z' \in X : \varphi_n(z, z') \to (Z(z) \leftrightarrow Z(z')) \tag{4}$$
$$\wedge\ Z(y) \tag{5}$$
$$\wedge\ B \cap (X_0 \backslash \{x'\}) = \emptyset \tag{6}$$
$$\wedge\ Y_0 \backslash (\lambda^{-1}(\Gamma) \cup \{y\}) \subseteq B \tag{7}$$
$$\wedge\ \forall z, z' \in Z : \varphi_n(z, z') \to (B(z) \leftrightarrow \neg B(z')) \tag{8}$$
$$\wedge\ \forall z_1, z_2, z_3, z_1', z_2', z_3' \in X :$$
$$(\varphi_n(z_1, z_1') \wedge \lambda(z_1) \notin \Gamma \wedge z_1 \lessdot z_2 \lessdot z_3 \wedge z_1' \lessdot z_2' \lessdot z_3' \wedge \neg Z(z_3))$$
$$\to ((B(z_1) \wedge \neg B(z_1')) \leftrightarrow (\neg B(z_3) \leftrightarrow B(z_3'))) \tag{9}$$
$$\wedge\ \forall z_1, z_2, z_3, z_1' \in X :$$
$$(\varphi_n(z_1, z_1') \wedge z_1 \lessdot z_2 \lessdot z_3 \wedge Z(z_3) \wedge B(z_1)) \to B(z_1') \tag{10}$$

Let $m \geqslant 1$ and $w \in L_m$. Furthermore, assume $X, X_0, Y_0, Z, B \subseteq \text{pos}(w)$ and $x', y' \in \text{pos}(w)$. Then (1) expresses $x \leqslant y$ and $X = \{x, x+1, \ldots, y\}$. By the induction hypothesis for φ_n, (2) says that x is even, $x' = x + 2F_n(m)$, $X_0 = \{x, x+1, \ldots, x+2F_n(m)\}$, and, because of $X_0 \subseteq X$, also $x' = x+2F_n(m) \leqslant y$. Similarly, (3) expresses $y' = y - 2F_n(m)$ and $Y_0 = \{y', y'+1, \ldots, y\}$. From (4), we obtain $Z \cap X = (x + 2F_n(m)\mathbb{N}) \cap X$ which, together with (5) ensures $y = x + 2kF_n(m)$ for some $k > 0$. In other words, the set Z divides the interval X into blocks of length $2F_n(m)$ each. The first block starts at position x and the last one at position y'. With any such block, we can associate a natural number depending on the set B: if the block starts at position $z \in Z$ and $H = \{i < F_n(m) \mid z + 2i \in B\}$, then the associated number is $\sum_{i \in H} 2^i$. In other words, we understand each block as a binary number (least significant bit first) where B contains those bits set to 1. Recalling that $X_0 \backslash \{x'\}$ is the first block, (6) expresses that its associated number is 0. Dually, using (7) we deduce that $\sum_{0 \leqslant i < F_n(m)} 2^i = 2^{F_n(m)} - 1$ is the number associated with the final block $Y_0 \backslash \{y\}$. We show that the blocks "count" from 0 to $2^{F_n(m)} - 1$. By (8) the least significant bits of consecutive blocks alternate. Consider (9). The premise expresses that z_1 and z_1' mark the same position i in consecutive blocks and that this position is not the last one. Then the conclusion says that the ith bit drops from 1 to 0 if and only if the $(i+1)$th bit changes. Hence, (9) expresses that the number associated with the following block is obtained by adding one modulo $2^{F_n(m)}$. The final formula (10) ensures that the last (most significant) bit never drops from 1 to 0. Hence, the number of blocks must be $2^{F_n(m)}$. Since each of them is of length $2F_n(m)$, we obtain $y = x + 2F_n(m)2^{F_n(m)} = x + 2F_{n+1}(m)$.

By induction, $\varphi_n \in \text{M}\Sigma_n^1(\Gamma, \lessdot, \prec)$. Note that this formula occurs in (2), (3), (4), (8), (9), and (10). At all these places, it occurs either positively under the existential quantification in the very first line, or negatively under an additional universal quantification. Hence, $\varphi_{n+1} \in \text{M}\Sigma_{n+1}^1(\Gamma, \lessdot, \prec)$ as required. □

We can now prove the main result of this section:

Proposition 4.5. *The set $C \subseteq \Sigma^*$ of encodings of computations of the Turing machine M can be defined with a sentence $\psi \in \text{M}\Pi_n^1(\Gamma, \lessdot, \prec)$, i.e., such that $C = \{w \in \Sigma^* \mid w \models \psi\}$.*

Proof. By Lemma 4.2, there is a formula $\psi_0 \in \text{FO}(\Gamma, \lessdot, \prec)$ which defines the language $L = \bigcup_{m \geqslant 1} L_m \subseteq (A\Gamma)^*A$. So below, we restrict our attention to words $w \in L_m$ for some $m \geqslant 1$. We use the abbreviations

$(\lambda_r(x) = \gamma) = (\exists x' : (x \lessdot x' \wedge \lambda(x') = \gamma))$ and
$(\lambda_\ell(x) = \gamma) = (\exists x' : (x' \lessdot x \wedge \lambda(x') = \gamma))$.

Consider the formula

$$\psi_1 = \exists x, y : \lambda_r(x) = \triangleright \wedge \lambda_\ell(y) = \triangleleft \wedge \neg\exists z : (z \lessdot x \vee y \lessdot z) \\ \wedge \forall x, y, z : (x \lessdot y \lessdot z) \to (\lambda(x) = \triangleleft \leftrightarrow \lambda(z) = \triangleright).$$

Then, $w \models \psi_1$ if and only if its projection $\pi_\Gamma(w)$ is in $(\triangleright(\Gamma\setminus\{\triangleright, \triangleleft\})^*\triangleleft)^+$. Next, we have to make sure that each factor in $\triangleright(\Gamma\setminus\{\triangleright, \triangleleft\})^*\triangleleft$ is of length $F_n(m)$. For this, we introduce for $n \geqslant 1$ a formula $\varphi_n^<(x, y)$ defined as $\varphi_n(x, y)$ in the proof of Lemma 4.4 except that (7) is replaced by its negation $Y_0\setminus\{y\} \not\subseteq \lambda^{-1}(\Gamma) \cup B$ so that the value associated with the last block is strictly less than $2^{F_{n-1}(m)} - 1$. Therefore, $w \models \varphi_n^<(x, y)$ if and only if x is even and $y = x + 2kF_{n-1}(m)$ for some $0 < k < 2^{F_{n-1}(m)}$. We define

$$\psi_2 = \forall x : \lambda_r(x) = \triangleright \to \exists x', X : (\varphi_{n-1}(x, x') \wedge \text{interval}(x, x', X) \wedge X \cap \lambda^{-1}(\triangleleft) = \emptyset) \\ \wedge \forall x, x', X, y, y', Y : \begin{pmatrix} \lambda_r(x) = \triangleright \wedge \varphi_n^<(x, x') \wedge \text{interval}(x, x', X) \\ \wedge\ \lambda_\ell(y) = \triangleleft \wedge \varphi_{n-1}(y', y) \wedge \text{interval}(y', y, Y) \end{pmatrix} \to X \cap Y \subseteq \{x, x'\} \\ \wedge \forall x, y : (\varphi_n(x, y) \wedge \lambda_r(x) = \triangleright) \to (\lambda_\ell(y) = \triangleleft)$$

and we show that a word $w \in L_m$ with $m \geqslant 1$ satisfies $\psi_1 \wedge \psi_2$ if and only if its projection $\pi_\Gamma(w)$ is in $(\Gamma^{F_n(m)} \cap \triangleright(\Gamma\setminus\{\triangleright, \triangleleft\})^*\triangleleft)^+$.

First, assume that $w \models \psi_1 \wedge \psi_2$ for some $w \in L_m$ with $m \geqslant 1$. From ψ_1, we already know that $\pi_\Gamma(w) \in (\triangleright(\Gamma\setminus\{\triangleright, \triangleleft\})^*\triangleleft)^+$ and we have to show that each factor is of length $F_n(m)$. So let $x \in \text{pos}(w)$ be such that $\lambda_r(x) = \triangleright$ and let $y \in \text{pos}(w)$ be minimal with $y > x$ and $\lambda_\ell(y) = \triangleleft$. By the first conjunct of ψ_2 we deduce that $y > x + 2F_{n-1}(m)$. Let k be maximal with $x + 2kF_{n-1}(m) \leqslant y$. We have seen that $k \geqslant 1$. Towards a contradiction, assume that $k < 2^{F_{n-1}(m)}$ and let $x' = x + 2kF_{n-1}(m)$ so that $\varphi_n^<(x, x')$ holds. Further, let $y' = y - 2F_{n-1}(m)$, $X = \{x, x + 1, \ldots, x'\}$ and $Y = \{y', y' + 1, \ldots, y\}$. Using the second conjunct of ψ_2 we get $y' \geqslant x'$ which contradicts the maximality of k. Therefore, $y \geqslant x + 2F_n(m)$. Finally, using the third conjunct of ψ_2, we get $y = x + 2F_n(m)$ as desired.

Conversely, let $w \in L_m$ be such that $\pi_\Gamma(w) \in (\Gamma^{F_n(m)} \cap \triangleright(\Gamma\setminus\{\triangleright, \triangleleft\})^*\triangleleft)^+$. We already know that $w \models \psi_1$. It is easy to see that w satisfies the first and last conjuncts of ψ_2. Finally, let $x, x', y, y' \in \text{pos}(w)$ and $X, Y \subseteq \text{pos}(w)$ satisfying the premise of the second conjunct of ψ_2. Then, $x' = x + 2kF_{n-1}(m)$ for some $0 < k < 2^{F_{n-1}(m)}$ and $X = \{x, x + 1, \ldots, x'\}$. Also, $y' = y - 2F_{n-1}(m)$ and $Y = \{y', y' + 1, \ldots, y\}$. Now, either $y \leqslant x$ and we get $X \cap Y \subseteq \{x\}$. Or else $y \geqslant x + 2F_n(m)$ and we obtain $X \cap Y \subseteq \{x'\}$. Therefore, $w \models \psi_2$ as required.

Next, we consider

$$\psi_3 = \forall x, y, X : (\lambda_r(x) = \triangleright \wedge \varphi_n(x, y) \wedge \text{interval}(x, y, X)) \to |X \cap \lambda^{-1}(Q)| = 1$$

so that $w \models \psi_1 \wedge \psi_2 \wedge \psi_3$ if and only if its projection $\pi_\Gamma(w)$ is in

$$\left(\Gamma^{F_n(m)} \cap (\triangleright(\Gamma_{\text{tape}}\setminus\{\triangleright, \triangleleft\})^* Q (\Gamma_{\text{tape}}\setminus\{\triangleright, \triangleleft\})^*\triangleleft)\right)^+.$$

The last formula is

$$\psi_4 = \forall x_0, \ldots, x_5, y_0, \ldots, y_5 : \left(\varphi_n(x_0, y_0) \wedge \bigwedge_{0 \leqslant i < 5} x_i \lessdot x_{i+1} \wedge y_i \lessdot y_{i+1}\right) \\ \to (\lambda(x_1), \lambda(x_3), \lambda(x_5), \lambda(y_1), \lambda(y_3), \lambda(y_5)) \in R$$

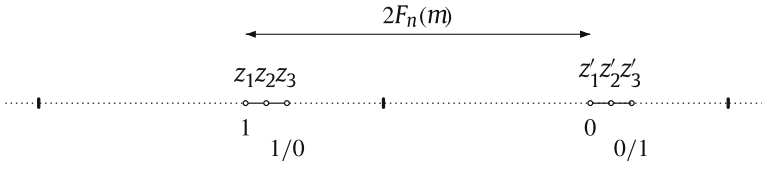

Fig. 4. Conjunct (9). This picture is a zoomed version of Fig. 3. The solid positions are consecutive positions from Z, z_1, z_2, and z_3 are consecutive positions and similarly for z'_1, z'_2, and z'_3 where the distance between z_1 and z'_1 equals $2F_n(m)$. Again, the bottom lines denote membership in B, one line for each side of the equivalence in (9).

Fig. 5. The traces $\eta(a)$ and $\eta(\gamma)$/.

Now, $w \models \bigwedge_{1\leqslant i\leqslant 4} \psi_i$ if and only if its projection is an m-computation. Therefore, the formula $\psi = \bigwedge_{0\leqslant i\leqslant 4} \psi_i$ defines the language C.

Moreover, $\psi \in \mathrm{M}\Pi^1_n(\Gamma, \lessdot, \prec)$. Indeed, ψ_0 and ψ_1 are first-order. In the first conjunct of ψ_2, the conclusion is in $\mathrm{M}\Sigma^1_{n-1}$ and is under a universal quantification. Finally, in all remaining conjuncts of ψ, the formulas $\varphi_n, \varphi_n^{\leq}$ and φ_{n-1} occur negatively under some universal quantifications. □

4.2. From words to traces

In this section, we will extend the infinite alphabet Σ to a dependence alphabet (Σ', D) such that letters from Σ are mutually dependent.[6] For a dependence clique $\Delta \subseteq \Sigma'$ and a trace $t \in \mathbb{M}(\Sigma, D)$, the Δ-labeled nodes form a chain in t and therefore define a word from Δ^* that we denote $\pi_\Delta(t)$. The main task will be the construction of a set of finite traces $C' \subseteq \mathbb{M}(\Sigma', D)$ definable in $\mathrm{M}\Pi^1_n(\mathsf{N}, \lessdot)$ such that $\pi_\Sigma(C') = C$, the language from Lemma 4.5. For the construction of C', we consider a disjoint copy $\overline{A} = \{\overline{a} \mid a \in A\}$ of A and we let

$$\Sigma' = \Sigma \uplus \overline{A} \uplus \{\dagger\} = \Gamma \uplus A \uplus \overline{A} \uplus \{\dagger\}.$$

The dependence relation is given by

$$D = (\Sigma \cup \{\dagger\})^2 \cup \{(a, \overline{a}), (\overline{a}, a), (\overline{a}, \overline{a}) \mid a \in A\}.$$

For simplicity, we write $\mathbb{M}$ for the trace monoid $\mathbb{M}(\Sigma', D)$.

Define a homomorphism $\eta : \Sigma^* \to \mathbb{M}$ by

$$\eta(\sigma) = \begin{cases} \overline{a}\, a\, \dagger\, \overline{a} & \text{if } \sigma = a \in A \\ \gamma\dagger & \text{if } \sigma = \gamma \in \Gamma. \end{cases}$$

The traces $\eta(a)$ and $\eta(\gamma)$ for $a \in A$ and $\gamma \in \Gamma$ are depicted in Fig. 5. Note that, for $\sigma \in \Sigma$ we have $\pi_\Sigma(\eta(\sigma)) = \sigma$. Since the letters from Σ are mutually dependent, we get $\pi_\Sigma(\eta(w)) = w$ for all words $w \in \Sigma^*$. Note also that for any $w \in \Sigma^*$, we have $\pi_{\Sigma\cup\{\dagger\}}(\eta(w)) \in (\Sigma\dagger)^*$. The language C' that we will define here is precisely $\dagger\eta(C)$ so that $\pi_\Sigma(C') = C$ as claimed.

As in the previous section, we will not allow formulae $\lambda(x) = \sigma$ for arbitrary letters $\sigma \in \Sigma'$ since we do not want our formulae to depend on $A \cup \overline{A}$. Hence, these atomic propositions are restricted to letters in $\Gamma \cup \{\dagger\} \subseteq \mathsf{N}$.

Lemma 4.6. *There is a formula $\varphi \in \mathrm{FO}(\mathsf{N}, \lessdot)$ such that for any trace $t \in \mathbb{M}$, we have $t \models \varphi$ iff $t \in \dagger\eta(\Sigma^*)$.*

Proof. We will define φ as a conjunction $\varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4$. The formula φ_1 will be satisfied by a trace $t \in \mathbb{M}$ iff $\pi_{\Sigma\cup\{\dagger\}}(t) \in \dagger(\Sigma\dagger)^*$. It is defined by

$$\begin{aligned} \varphi_1 = \ & \exists x\, (x \text{ minimal} \wedge \lambda(x) = \dagger) \\ & \wedge \forall x\, (\lambda(x) = \dagger \to (x \text{ maximal} \vee \exists y \exists z\, (x \lessdot y \lessdot z \wedge \lambda(y) \neq \dagger \wedge \lambda(z) = \dagger))). \end{aligned}$$

Let $t = (V, \leq, \lambda) \in \mathbb{M}$ be a trace. If $t \models \varphi_1$, then t contains a minimal node that is $\dagger$-labeled, and from any non-maximal $\dagger$-labeled node, we reach another one in just two $\lessdot$-steps. Hence, t contains a maximal $\lessdot$-chain labeled in $\dagger((\Sigma'\backslash\{\dagger\})\dagger)^*$. Since consecutive nodes in a maximal $\lessdot$-chain carry dependent letters, this chain actually belongs to $\dagger(\Sigma\dagger)^*$. Since $\Sigma \cup \{\dagger\}$ forms a dependence clique, all $(\Sigma \cup \{\dagger\})$-labeled nodes must be in the chain. We deduce that $\pi_{\Sigma\cup\{\dagger\}}(t) \in \dagger(\Sigma\dagger)^*$.

Let, conversely, $\pi_{\Sigma\cup\{\dagger\}}(t) \in \dagger(\Sigma\dagger)^*$. Let x be the minimal $\dagger$-labeled node of t. Since the projection starts with $\dagger$, this node x does not dominate any Σ-labeled node. Since only letters from $\Sigma \cup \{\dagger\}$ are dependent from $\dagger$, the node x is minimal in t. Now let x and z be two consecutive $\dagger$-labeled nodes of t. Then, in between them, there is a unique node y with $\lambda(y) \in \Sigma$. Since all neighbors of x and z have to carry labels in $\Sigma \cup \{\dagger\}$, this implies $x \lessdot y \lessdot z$. Since the last letter of the projection is $\dagger$, we showed that $t \models \varphi_1$.

We restrict our attention below to traces t that satisfy φ_1, i.e., such that $\pi_{\Sigma\cup\{\dagger\}}(t) \in \dagger(\Sigma\dagger)^*$. In particular, a node x of t is labeled in Σ if and only if $\exists y\, (x \lessdot y \wedge \lambda(y) = \dagger)$. We will simply write $\lambda(x) \in \Sigma$ for this formula. We also use the abbreviations $\lambda(x) \in A$ for $\lambda(x) \in \Sigma\backslash\Gamma$, and $\lambda(x) \in \overline{A}$ for $\lambda(x) \notin \Sigma \cup \{\dagger\}$.

[6] Since Σ is infinite, also the set Σ' is infinite. Although we only defined traces over finite dependence alphabets, the definitions go through for infinite ones as well.

The formula φ_2 will ensure that any Σ-labeled node is the center of some factor $\eta(\sigma)$. For Γ-labeled nodes, this is already implied by φ_1. For $\sigma \in A$, it turns out to be sufficient to require the existence of at least two upper and two lower neighbors. Formally we define

$$\varphi_2 = \forall y\, (\lambda(y) \in A \to \exists x \exists z : x \lessdot y \lessdot z \wedge \lambda(x) \neq \dagger \wedge \lambda(z) \neq \dagger).$$

Formula φ_2 expresses that any a-labeled node y (with $a \in A$) has at least one lower and one upper neighbor x, z that are not labeled $\dagger$. Because of the structure of the dependence relation D, the only possibility is that $\lambda(y) = \lambda(z) = \overline{a}$. Note that $D(\overline{a}) \subseteq D(a)$. Hence, y is the only upper neighbor of x and the only lower neighbor of z. Thus, the neighborhood of y excluding the $\dagger$-labeled lower neighbor forms a factor of the form $\eta(a)$.

Next, we express that any node in the trace t belongs to one of the factors $\eta(\sigma)$ with $\sigma \in \Sigma$:

$$\varphi_3 = \forall y\, (\lambda(y) \in \overline{A} \to \exists z\, (\lambda(z) \in A \wedge (y \lessdot z \vee z \lessdot y))).$$

It remains to express that the factors considered above are mutually disjoint:

$$\varphi_4 = \forall y\, (\lambda(y) \in \overline{A} \to \neg \exists x \exists z : (x \lessdot y \lessdot z \wedge \lambda(x) \in A \wedge \lambda(z) \in A)).$$

The only possibility for a node y of t to belong to two factors is that $\lambda(y) = \overline{a}$ for some $a \in A$ and the two factors are of the form $\eta(a)$. But then y would have two a-labeled neighbors x and z – this is excluded by formula φ_4.

Let $\dagger\sigma_1\dagger\sigma_2\dagger \ldots \sigma_n\dagger$ be the projection of t to $\Sigma \cup \{\dagger\}$. Then, by what we showed so far, we deduce that $t = \dagger\eta(\sigma_1)\eta(\sigma_2)\ldots\eta(\sigma_n)$. $\square$

For any word $w \in \Sigma^*$ we have $w = \pi_\Sigma(\dagger\eta(w))$. Thus, the word w can be seen as a chain in the trace $\dagger\eta(w)$. Note that the predicate $\lambda(x) \in \overline{A}$ can be expressed in $\dagger\eta(w)$ by $\lambda(x) \neq \dagger \wedge \neg\exists y\, (x \lessdot y \wedge \lambda(y) = \dagger)$. We use it as a macro below. We will next prove that the relations $\lessdot$ and $\prec$ of w can be expressed by first-order formulas in $\dagger\eta(w)$. To this aim, we define

$$\begin{aligned} \mathrm{cover}(x, y) &= \exists z\, (\lambda(z) = \dagger \wedge x \lessdot z \lessdot y), \\ \mathrm{nx}(x, y) &= \exists x' \exists y'\, (\lambda(x') \in \overline{A} \wedge \lambda(y') \in \overline{A} \wedge x \lessdot x' \lessdot y' \lessdot y). \end{aligned}$$

Lemma 4.7. *Let $w \in \Sigma^*$ and $t = \dagger\eta(w) = (V, \leq, \lambda)$. Suppose furthermore $x, y \in V$ with $\lambda(x), \lambda(y) \in \Sigma$. Then we have*

1. *$w \models x \lessdot y$ iff $t \models \mathrm{cover}(x, y)$.*
2. *$w \models x \prec y$ iff $t \models \mathrm{nx}(x, y)$.*

Proof. Let $w = a_1 a_2 \ldots a_n$ with $a_i \in \Sigma$. Note that those nodes that are labelled in $\Sigma \cup \{\dagger\}$ form a maximal chain in t corresponding to a word in $\dagger(\Sigma\dagger)^*$. This ensures the first statement.

Suppose $x \prec y$ in the word w. Then, by the definition of $\prec$, we have $\lambda(x) = \lambda(y) = a \in A$, $x < y$, and there is no z with $x < z < y$ and $a = \lambda(z)$. The definition of η implies the existence of x' and y' with $x \lessdot x'$, $y' \lessdot y$, and $\lambda(x') = \lambda(y') = \overline{a}$. Since no a occurs in between x and y, we obtain $x' \lessdot y'$. Thus, $t \models \mathrm{nx}(x, y)$.

Conversely, suppose $t \models \mathrm{nx}(x, y)$. Then there are x' and y' with $x \lessdot x' \lessdot y' \lessdot y$ and $\lambda(x'), \lambda(y') \in \overline{A}$. Since $(\lambda(x'), \lambda(y')) \in D$, the construction of η ensures $\lambda(x') = \lambda(y') = \overline{a}$ for some $a \in A$. For the same reason, we obtain $\lambda(x) = a = \lambda(y)$ and there cannot be a further occurrence of a in between x and y. Hence, $x \prec y$ in the word w. $\square$

This allows immediately to derive the following consequence since C is definable in $\mathrm{M}\Pi^1_n(\Gamma, \lessdot, \prec)$:

Proposition 4.8. *The language $\dagger\eta(C)$ is $\mathrm{M}\Pi^1_n(\mathrm{N}, \lessdot)$-definable, i.e., there is a sentence $\psi' \in \mathrm{M}\Pi^1_n(\mathrm{N}, \lessdot)$ such that $C' = \dagger\eta(C) = \{t \in \mathbb{M} \mid t \models \psi'\}$.*

Proof. By Lemma 4.5, there is a sentence ψ in $\mathrm{M}\Pi^1_n(\Gamma, \lessdot, \prec)$ such that $C = \{w \in \Sigma^* \mid w \models \psi\}$. For $\xi \in \mathrm{MSO}(\Gamma, \lessdot, \prec)$, we construct recursively $\overline{\xi}$ as follows:

$$\begin{aligned} \overline{(\lambda(x) = e)} &= (\lambda(x) = e) \\ \overline{x \lessdot y} &= \mathrm{cover}(x, y) \\ \overline{x \prec y} &= \mathrm{nx}(x, y) \\ \overline{\neg\varphi} &= \neg\overline{\varphi} \\ \overline{\varphi \vee \psi} &= \overline{\varphi} \vee \overline{\psi} \\ \overline{\exists x \varphi} &= \exists x\, (\overline{\varphi} \wedge \lambda(x) \in \Sigma) \\ \overline{\exists X \varphi} &= \exists X\, (\overline{\varphi} \wedge \forall x\, (x \in X \to \lambda(x) \in \Sigma)) \end{aligned}$$

Then, by Lemma 4.7, we deduce that for $w \in \Sigma^*$ we have $w \models \xi$ iff $\dagger\eta(w) \models \overline{\xi}$. Then, we let $\psi' = \varphi \wedge \overline{\psi}$ where φ is the FO(N, $\lessdot$)-sentence from Lemma 4.6. Now the result follows immediately from Lemmas 4.5–4.7. □

4.3. The lower bound

Now we can prove the main theorem of this section.

Proof of Theorem 4.1. Recall that the deterministic Turing machine M works in space $F_n(m) - 3$ where m is the length of the input word.

Consider the $\mathrm{M}\Pi^1_n(\mathrm{N}, \lessdot)$-definable temporal logic TL_n based on the modality SU, the usual boolean connectives, and the constant COMPUTATION with $[\![\mathrm{COMPUTATION}]\!] = \psi'$, the formula from Proposition 4.8 defining $\dagger\eta(C)$.

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

$$\mathrm{INIT}_v = \neg\Gamma \ \mathsf{SU}\ (\triangleright \wedge \neg\Gamma \ \mathsf{SU}\ (q_0 \wedge \neg\Gamma \ \mathsf{SU}\ (v_1 \wedge \cdots \neg\Gamma \ \mathsf{SU}\ (v_m \wedge (\neg\Gamma \vee \Box) \ \mathsf{SU} \triangleleft) \cdots)))$$

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 $\Sigma_m = A_m \cup \Gamma \subseteq \Sigma$ and $\Sigma'_m = \Sigma_m \cup \{\dagger\} \cup \overline{A_m}$ where $|A_m| = m$, $\overline{A_m} = \{\overline{a} \mid a \in A_m\}$ and the dependence relation D defined as in Section 4.2. 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 $\mathrm{COMPUTATION} \wedge \mathrm{INIT}_v \wedge \top \ \mathsf{SU}\ q_1$. Note that this formula can be constructed from v in linear time. Therefore, the uniform satisfiability problem for TL_n is **n-EXPSPACE**-hard. □

Remark 4.9. Note that, apart from the boolean connectives, the logic TL_n contains only the constant COMPUTATION and the binary modality SU. In our hardness proof, the binary SU 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 with these three unary filter modalities in the style of [9]. Furthermore, the temporal logic could be deprived of constant formulas a for $a \notin \Gamma \cup \{\dagger\}$ since they are not used in the hardness proof.

References

[1] B. Adsul, M. Sohoni, Complete and tractable local linear time temporal logics over traces, in: Proceedings of the ICALP'02, LNCS, vol. 2380, Springer, Berlin, 2002, pp. 926–937.
[2] R. Alur, D. Peled, W. Penczek, Model-checking of causality properties, in: Proceedings of the LICS'95, IEEE Computer Society Press, Silver Spring, MD, 1995, pp. 90–100.
[3] V. Diekert, P. Gastin, Local temporal logic is expressively complete for cograph dependence alphabets, Information and Computation 195 (2004) 30–52., a preliminary version appeared at LPAR'01, LNAI 2250, Springer, Berlin, pp. 55–69.
[4] V. Diekert, G. Rozenberg (Eds.), The Book of Traces, World Scientific, Singapore, 1995.
[5] V. Diekert, P. Gastin, Pure future local temporal logics are expressively complete for Mazurkiewicz traces, in: Proceedings of the LATIN'04, LNCS, vol. 2976, Springer, Berlin, 2004, pp. 232–241.
[6] H.-D. Ebbinghaus, J. Flum, Finite Model Theory, Springer, Berlin, 1991.
[7] D. Gabbay, I. Hodkinson, M. Reynolds, Temporal Logic, Oxford University Press, Oxford, 1994.
[8] P. Gastin, D. Kuske, Uniform satisfiability in PSPACE for local temporal logics over Mazurkiewicz traces, Fundamenta Informaticae 80 (1–3) (2007) 169–197.
[9] P. Gastin, M. Mukund, An elementary expressively complete temporal logic for Mazurkiewicz traces, in: Proceedings of the ICALP'02, LNCS, vol. 2380, Springer, Berlin, 2002, pp. 938–949.
[10] P. Gastin, D. Kuske, Satisfiability and model checking for MSO-definable temporal logics are in PSPACE, in: Proceedings of the CONCUR'03, LNCS, vol. 2761, Springer, Berlin, 2003, pp. 222–236.
[11] P. Gastin, M. Mukund, K. Narayan Kumar, Local LTL with past constants is expressively complete for Mazurkiewicz traces, in: Proceedings of the MFCS'03, LNCS, vol. 2747, Springer, Berlin, 2003, pp. 429–438.
[12] P. Gastin, D. Kuske, Uniform satisfiability problem for local temporal logics over Mazurkiewicz traces, in: Proceedings of the CONCUR'05, LNCS, vol. 3653, Springer, Berlin, 2005, pp. 533–547.
[13] H.J. Keisler, W.B. Lotfallah, Shrinking games and local formulas, Annals of Pure and Applied Logic 128 (1–3) (2004) 215–225.
[14] Y. Kesten, A. Pnueli, L. Raviv, Algorithmic verification of linear temporal logic specifications, in: Proceedings of the ICALP'98, LNCS, vol. 1443, Springer, Berlin, 1998, pp. 1–16.
[15] O. Kupferman, M. Vardi, Weak alternating automata are not that weak, ACM Transaction on Computational Logic 2 (3) (2001) 408–429.
[16] O. Matz, Dot-depth and monadic quantifier alternation over pictures, Ph.D. Thesis, RWTH Aachen, 1999.
[17] O. Matz, N. Schweikardt, W. Thomas, The monadic quantifier alternation hierarchy over grids and graphs, Information and Computation 179 (2) (2002) 356–383.
[18] S. Miyano, T. Hayashi, Alternating finite automata on ω-words, Theoretical Computer Science 32 (1984) 321–330.
[19] M. Mukund, P. Thiagarajan, Linear time temporal logics over Mazurkiewicz traces, in: Proceedings of the MFCS'96, LNCS, vol. 13, Springer, Berlin, 1996, pp. 62–92.
[20] D.E. Muller, A. Saoudi, P. E. Schupp, Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time, in: Proceedings of the LICS'88, IEEE Computer Society Press, Silver Spring, MD, 1988, pp. 422–427.
[21] T. Schwentick, K. Bartelmann, Local normal forms for first-order logic with applications to games and automata, Discrete Mathematics and Computer Science 3 (1999) 109–124.
[22] P. Thiagarajan, A trace based extension of linear time temporal logic, in: Proceedings of the LICS'94, IEEE Computer Society Press, Silver Spring, MD, 1994, pp. 438–447.
[23] P. Thiagarajan, A trace consistent subset of PTL, in: Proceedings of the CONCUR'95, LNCS, vol. 962, Springer, Berlin, 1995, pp. 438–452.
[24] I. Walukiewicz, Difficult configurations-on the complexity of LTrL, Formal Methods in System Design 26 (1) (2005) 27–43.