Temporal logics over linear time domains are in PSPACE

Alexander Rabinovich

The Blavatnik School of Computer Science
Tel Aviv University, Tel Aviv, Israel 69978
e-mail: rabinoa@post.tau.ac.il

Abstract. We investigate the complexity of the satisfiability problem of temporal logics with a finite set of modalities definable in the existential fragment of monadic second-order logic. We show that the problem is in PSPACE over the class of all linear orders. The same techniques show that the problem is in PSPACE over many interesting classes of linear orders.

1 Introduction

A major result concerning linear-time temporal logics is Kamp's theorem [12, 9] which says that $TL(\mathsf{Until}, \mathsf{Since})$, the temporal logic having Until and Since as the only modalities, is expressively complete for first-order monadic logic of order over the class of Dedekind complete linear orders.

The order of natural numbers $\omega = (\mathbb{N}, <)$ and the order of the real numbers $(\mathbb{R}, <)$ are both Dedekind-complete. Another important class of Dedekind-complete orders is the class of ordinals. However, the order of the rationals is not Dedekind-complete. Stavi introduced two modalities Until_{Stavi} and Since_{Stavi} and proved that the temporal logic having the four modalities Until, Since, Until_{Stavi} and Since_{Stavi} is expressively complete for first-order monadic logic of order over the class of all linear orders [9].

Our concern in this paper will be with the complexity of the satisfiability problem for temporal logics over various classes of linear orders.

Sistla and Clarke [21] proved that the satisfiability problem for $TL(\mathsf{Until}, \mathsf{Since})$ over ω-models is PSPACE-complete. In [7], it was proven that the satisfiability problem for $TL(\mathsf{Until}, \mathsf{Since})$ over the class of all ordinals is PSPACE-complete. Cristau [6] provided a double exponential space algorithm for the satisfiability of the temporal logic having the four modalities Until, Since, Until_{Stavi} and Since_{Stavi} over the class of all linear orders. These proofs are based on automata theoretical techniques.

Burgess and Gurevich [5] proved that $TL(\mathsf{Until}, \mathsf{Since})$ is decidable over the reals. They provided two proofs. The first involves an indirect reduction to Rabin's theorem on the decidability of the monadic second-order logic over the full binary tree [14]. The second one is based on the model-theoretical composition method. Both proofs provide algorithms of non-elementary complexity.

Reynolds [17, 16] proved that the satisfiability problem for $TL(\mathsf{Until}, \mathsf{Since})$ over the reals is PSPACE-complete and that the temporal logic with only the Until modality is PSPACE-complete over the class of all linear orders. The proofs in [17, 16] use temporal mosaics and are very non-trivial and difficult to grasp. Reynolds conjectured [16] that the satisfiability problem for the logic with Stavi's modalities over the class of all linear orders is in PSPACE. Our results imply this conjecture.

Let TL be a temporal logic with a finite set of modalities definable in the existential fragment of monadic second-order logic. We prove in a uniform manner that the satisfiability problem for TL is in PSPACE over the following classes of time domains: (1) all linear orders, (2) ordinals, (3) scattered linear orders, (4) Dedekind-complete linear orders, (5) continuous orders, (6) rationals, (7) reals. The proofs are based both on the composition method and on automata theoretical techniques and are easily adapted to various classes of structures and temporal and modal logics.

Our first reduction uses the following notion. Let $\varphi(X_1, \ldots, X_k)$ be a formula with free set variables among $X_1, \ldots, X_k$. An instance of φ is a formula obtained by replacing $X_1, \ldots, X_k$ by monadic predicate names. Let Φ be a set of formulas. A Φ-conjunctive formula is a conjunction of instances of formulas from Φ.

Our first reduction shows that for every temporal logic $\mathcal{L}$ with a finite set of modalities definable in the existential fragment of monadic second-order logic there is a finite set Φ of first-order formulas and a linear time algorithm that reduces the satisfiability problem for $\mathcal{L}$ to the satisfiability problem for Φ-conjunctive formulas. This algorithm is based on a simple unnesting procedure and works as it is for a much broader class of modal logics.

Next, we introduce recursively definable classes of structures. Our second reduction shows that for every finite set Φ of first-order formulas and every recursively definable class of structures $\mathcal{C}$ the satisfiability problem for the Φ-conjunctive formulas over $\mathcal{C}$ is in EXPTIME. Like the first reduction, this reduction is quite general; it relies on the composition method and is sound not only for linear orders. The first two reductions give an almost free EXPTIME algorithm for many temporal and modal logics with finite sets of modalities.

To obtain PSPACE upper bound we need more subtle arguments. We assign a rank to every structure in a recursively definable class. An algorithm similar to the algorithm in the second reduction shows that for every polynomial p the problem whether a Φ-conjunctive formula φ is satisfiable over the structures of rank $p(|\varphi|)$ is in PSPACE. The main effort to show that the satisfiability problem for a recursively definable class is in PSPACE is to establish that a formula is satisfiable if it is satisfiable over the structures of a polynomial rank in the size of the formula. We prove such a bound for many interesting classes of linear orders. Our proof uses an automata-theoretical characterization of the temporal logic with Stavi's modalities over the linear orders found by Cristau [6].

The paper is organized as follows. The next section recalls basic definitions about monadic second-order logic, its fragments and temporal logics. Sect. 3 states a linear reduction from temporal logics to conjunctive formulas. Sect. 4

reviews basic notions about the compositional method. Sect. 5 introduces recursively defined classes of structures and Sect. 6 presents an exponential algorithm for the satisfiability of conjunctive formulas over these classes. Sect. 7 presents a PSPACE algorithm for the satisfiability of conjunctive formulas over the class of all linear orders and states a key lemma needed for its complexity analysis. Sect. 8 introduces finite base automata over arbitrary linear orders. Sect. 9 proves the main lemma about runs of automata which is needed for the proof of PSPACE bound of our algorithm. Sect. 10 proves in a "plug-and-play" manner PSPACE bound over several interesting classes of linear orders and discusses related works.

Detailed proofs can be found in [15].

2 Monadic Logics and Temporal Logics

2.1 Monadic second-order logic

Monadic second-order logic (MSO) is the fragment of the full second-order logic allowing quantification only over elements and monadic predicates. One way to define the monadic second-order language for a signature Δ (notation MSO(Δ)) is to augment the first-order language for Δ by quantifiable monadic predicate variables (set variables) and by new atomic formulas $X(t)$, where t is a first-order variable and X is a monadic predicate variable. The monadic predicate variables range over all subsets of a structure for Δ.

The *quantifier depth* of a formula φ is defined as usual and is denoted by qd(φ).

We will use lower case letters t, t' for the first-order variables and upper case letters X, Y, Z for the monadic variables.

An MSO formula is existential if it is of the form $\exists X_1 \dots \exists X_n \varphi$, where φ does not contain second-order quantifiers. The existential fragment of MSO consists of existential MSO formula and is denoted by ∃-MSO.

The first-order fragment of MSO contains formulas without the second-order quantifiers. These formulas might contain free second-order variables which play the same role as monadic predicate names. Hence, a formula in this fragment is interpreted over an expansion of Δ structures by predicates which provide meaning for the monadic variables. Sometimes, these free variables will serve as metavariables. If $\varphi(X_1, X_2)$ is a formula and P, Q are monadic predicate names, we will say that the formula obtained from φ by replacing X_1 by P and X_2 by Q is an *instance* of φ.

2.2 Temporal Logics and Truth Tables

Temporal logics use logical constructs called *"modalities"* to create a language free from quantifiers. Below is the general logical framework to define temporal logics:

The syntax of the Temporal Logic $TL(O_1^{(k_1)}, \dots, O_n^{(k_n)})$ has in its vocabulary *monadic predicate variables* $X_1, X_2, \dots$ and a sequence of *modality names* with a prescribed arity, $O_1^{(k_1)}, \ldots, O_n^{(k_n)}$ (the arity notation is usually omitted). The formulas of this temporal logic are given by the grammar:

$$\varphi \ ::= \ X \mid \neg\varphi \mid \varphi \wedge \varphi \mid O^{(k)}(\varphi_1, \cdots, \varphi_k)$$

When particular modality names are unimportant or are clear from the context, we omit them and write TL instead of $TL(O_1^{(k_1)}, \ldots, O_n^{(k_n)})$.

Structures for TL are partial orders with monadic predicates $\mathcal{M} = \langle A, <, P_1, P_2, \ldots, P_n, \ldots \rangle$, where the predicate P_i is assigned to a predicate variable X_i. Every modality $O^{(k)}$ is interpreted in every structure $\mathcal{M}$ as an operator $O_{\mathcal{M}}^{(k)} : [\mathcal{P}(A)]^k \rightarrow \mathcal{P}(A)$ which assigns "the set of points where $O^{(k)}[S_1 \ldots S_k]$ holds" to the k-tuple $\langle S_1 \ldots S_k \rangle \in \mathcal{P}(A)^k$. (Here, $\mathcal{P}$ is the power set notation, and $\mathcal{P}(A)$ denotes the set of all subsets of the domain A of $\mathcal{M}$.) Once every modality corresponds to an operator, the relation "φ holds in $\mathcal{M}$ at an element a" (notations $\langle \mathcal{M}, a \rangle \models \varphi$) is defined as follows:

- for atomic formulas $\langle \mathcal{M}, a \rangle \models X$ iff $a \in P$, where the monadic predicate P is assigned to X.
- for Boolean combinations the definition is the usual one.
- for modalities: $\langle \mathcal{M}, a \rangle \models O^{(k)}(\varphi_1, \cdots, \varphi_k)$ iff $a \in O_{\mathcal{M}}^{(k)}(P_{\varphi_1}, \cdots, P_{\varphi_k})$, where $P_\varphi = \{ b \mid \langle \mathcal{M}, b \rangle \models \varphi \}$.

Usually, we are interested in a more restricted case; for the modality to be of interest the operator $O^{(k)}$ should reflect some intended connection between the sets A_{φ_i} of points satisfying φ_i and the set of points $O[A_{\varphi_1}, \ldots, A_{\varphi_k}]$. The intended meaning is usually given by a formula in an appropriate predicate logic.

Truth Tables: A formula $\overline{O}(t_0, X_1, \ldots X_k)$ in the predicate logic L is a *Truth Table* for the modality O if for every structure $\mathcal{M}$ and subsets $P_1, \ldots, P_k$ of $\mathcal{M}$

$$O_{\mathcal{M}}(P_1, \ldots, P_k) = \{a : \ \mathcal{M} \models \overline{O}[a, P_1, \ldots, P_k]\} \ .$$

Thus, the modality $\Diamond X$, "*eventually* X", is defined by

$$\varphi(t_0, X) \equiv \exists t > t_0 (t \in X).$$

The modality $X \mathsf{Until}\, Y$, "X *strict until* Y", is defined by

$$\exists t_1 (t_0 < t_1 \wedge t_1 \in Y \wedge \forall t (t_0 < t < t_1 \rightarrow t \in X)).$$

A truth table $\varphi(t, Y_1, \cdots, Y_k)$ defines in every structure a function from k-tuples of subsets. It associates with the tuple $Y_1, \cdots, Y_k$ of subsets of a structure $\mathcal{M}$, the set of elements t in $\mathcal{M}$ that satisfy $\varphi(t, Y_1, \cdots, Y_k)$ in $\mathcal{M}$. This is a special case of a more general way to define a function on all the structures in a given class of structures. Here is the formal notion of a definable functional.

Definition 2.1 *1. Let L be a first-order or monadic second-order logic language, and let $\mathcal{M}$ be a structure. Let $\varphi(X, Y_1, \cdots, Y_k)$ be a formula in L with no free first-order variables, and with no set variables except for those specified. φ is an* implicit definition of the functional $X = f_\varphi^{\mathcal{M}}(Y_1, \cdots, Y_k)$ *if for any k subsets $Y_1, \cdots, Y_k$ of $\mathcal{M}$, X is the only subset of $\mathcal{M}$ for which $\mathcal{M} \models \varphi(X, Y_1, \cdots, Y_k)$.*

2. *A modality* $\mathcal{O}(Y_1, \cdots, Y_k)$ *of a temporal logic has a* generalized truth table $\varphi(X, Y_1, \cdots, Y_k)$ *in a structure* $\mathcal{M}$ *if* φ *implicitly defines the operator of* $\mathcal{O}$*; i.e., given subsets* $Y_1, \cdots, Y_k$ *of a structure* $\mathcal{M}$*,*

$$\langle \mathcal{M}, a\rangle \models \mathcal{O}(Y_1, \cdots, Y_k) \quad \text{iff} \quad a \in f_\varphi^{\mathcal{M}}(Y_1, \cdots, Y_k).$$

φ *is a generalized truth table for* $\mathcal{O}$ *in a class* $\mathcal{C}$ *of structures if* φ *is a generalized truth table for* $\mathcal{O}$ *in every* $\mathcal{M} \in \mathcal{C}$*.*

If the logic is a second-order logic, then this definition is a special case of the classical definition of a function defined by a formula. Note that if $\theta(t_0, Y_1, \cdots, Y_k)$ is a truth table for a modality $\mathcal{O}$, then $\forall t[X(t) \leftrightarrow \theta(t, Y_1, \cdots, Y_k)]$ is a generalized truth table for $\mathcal{O}$. Therefore, the notion of a generalized truth table is more general than that of a truth table. It is strictly more general. For example, it is well-known that there is no first-order formula $\varphi(t, X)$ which defines over the naturals the set of points preceded by an even number of points in X; however, it is easy to write a first-order formula $\psi(Y, X)$ which defines this modality over $(\mathbb{N}, <)$.

If a modality O has a generalized truth table $\varphi(X, Y_1, \cdots, Y_k)$, where φ is an existential monadic second-order formula, then $\exists X\big((X(t_0)) \wedge \varphi\big)$ is an $\exists$-MSO truth table for O. Hence, a modality has an $\exists$-MSO truth table iff it has an $\exists$-MSO generalized truth table and we will say that it is $\exists$-MSO definable.

There are $\exists$-MSO definable modalities which are not definable even by generalized truth tables of the first-order logic. For example, there is an $\exists$-MSO formula $\varphi(Y, X)$ that expresses "Y holds at t if $\neg X(t)$ and t precedes by a block of X of length $3m$ some $m > 0$", i.e., $X(t-1)$, $X(t-2), \ldots X(t-3m)$ and $\neg X(t-3m-1)$. However, there is no first-order formula equivalent to φ over $(\mathbb{N}, <)$.

Modal logics Temporal logics are examples of modal logics. The syntax of modal logics is defined exactly like the syntax of temporal logics. However, modal logics can be interpreted not only over linear or partial orders, but over structures of a more general signature Δ. Every modality $O^{(k)}$ is interpreted in every Δ-structure $\mathcal{M}$ as an operator $O^{(k)}_{\mathcal{M}} : [\mathcal{P}(\mathcal{M})]^k \to \mathcal{P}(\mathcal{M})$. Generalized truth tables are defined by formulas over Δ. We state our results for temporal logics; however, they hold for more general modal logics as well.

3 From Temporal Logic to Conjunctive Formulas

Let $\varphi(X_1, \ldots, X_k)$ be a formula with free set variables among $X_1, \ldots, X_k$. An *instance* of φ is a formula obtained by replacing $X_1, \ldots, X_k$ by monadic predicate names or monadic variables. Let Φ be a set of formulas. A *Φ-conjunctive* formula is a conjunction of instances of formulas from Φ.

Our first reduction shows that for every temporal logic $\mathcal{L}$ with a finite set of $\exists$-MSO definable modalities there is a finite set Φ of first-order formulas and a linear time algorithm that reduces the satisfiability problem for $\mathcal{L}$ to the satisfiability problem for Φ-conjunctive formulas.

Proposition 3.1 *Let TL be a temporal logic with a finite set of modalities. Assume that every modality of TL is ∃-MSO definable. Then there is a finite set Φ of first-order formulas, and a linear time algorithm which for every formula $\varphi(P_1, \ldots, P_m) \in TL$ computes a Φ-conjunctive formula $\psi(P_1, \ldots, P_m, Q_1, \ldots, Q_s)$ such that for every structure $\mathcal{M}$ in the signature $\{<, P_1, \ldots, P_m\}$, φ is satisfiable in $\mathcal{M}$ iff ψ is satisfiable in an expansion of $\mathcal{M}$ by monadic predicates (which are the interpretations of $Q_1, \ldots, Q_s$).*

The proof of this proposition is based on a simple unnesting procedure. A similar proposition holds for modal logics.

4 Elements of the Composition Method

Our proofs make use of a technique known as the composition method [8, 20, 11, 22]. To fix notations and to aid a reader unfamiliar with this technique, we briefly review the required definitions and results.

4.1 Hintikka formulas and *n*-types

Let $\mathcal{M}$ and $\mathcal{M}'$ be structures over a relational signature Σ. For $n \in \mathbb{N}$, the structures $\mathcal{M}$ and $\mathcal{M}'$ are said to be $\equiv^n$-equivalent if no first-order sentence of quantifier depth $\leq n$ distinguishes between $\mathcal{M}$ and $\mathcal{M}'$; i.e., for every φ of quantifier depth $\leq n$:

$$\mathcal{M} \models \varphi \text{ iff } \mathcal{M}' \models \varphi.$$

Lemma 4.1 (Hintikka Lemma) *For $n \in \mathbb{N}$ and a finite relational signature Σ we can compute a* finite *set $Hin^n := Hin^n(\Sigma)$ of sentences of quantifier depth $\leq n$ such that:*

1. *For every $\equiv^n$-equivalence class E there is a* unique *$\tau \in Hin^n$ such that for every Σ-structure $\mathcal{M}$: $\mathcal{M}, \in E$ if and only if $\mathcal{M} \models \tau$.*
2. *Every sentence with $\mathrm{qd}(\varphi) \leq n$ is equivalent to a (finite) disjunction of sentences from Hin^n. There is an algorithm which for every sentence φ computes a finite set $G_\varphi \subseteq Hin^{\mathrm{qd}(\varphi)}$ such that φ is equivalent to the disjunction of all the sentences from G_φ. Moreover, $\tau \in G_\varphi$ iff $\tau \rightarrow \varphi$.*

(Note that this general method to deal with sentences is not efficient in the sense of complexity theory, and that the algorithm is non-elementary.)

We call any member of Hin^n a *n-Hintikka sentence*. We use τ, τ_i, τ' to range over the Hintikka sentences.

Definition 4.2 (*n*-Type) *For $n \in \mathbb{N}$ and a Σ-structure $\mathcal{M}$, we denote by $type^n(\mathcal{M})$ the unique member of Hin^n satisfied in $\mathcal{M}$.*

4.2 The ordered sum of chains and of n-types

A (labeled) chain $\mathcal{M}$ is a linear order expanded by monadic predicates; if $\overline{P}$ is a set of monadic predicate names, and the signature of $\mathcal{M}$ is $\{<, \overline{P}\}$, we say $\mathcal{M}$ is a $\overline{P}$-chain. The *concatenation* or *ordered sum* of chains is defined as follows:

Definition 4.3 (Sum of Chains) *Let $\mathcal{I} := (I, <^{\mathcal{I}})$ be a linear order, $l \in \mathbb{N}$, and $\mathfrak{S} := (\mathcal{M}_\alpha \mid \alpha \in I)$ be a sequence of chains, where $\mathcal{M}_\alpha := (A_\alpha, <^\alpha, {P_1}^\alpha, \dots, {P_l}^\alpha)$. Assume that $A_\alpha \cap A_\beta = \emptyset$ whenever $\alpha \neq \beta$ are in I. The* ordered sum *of $\mathfrak{S}$ is the chain*

$$\sum_{\alpha \in \mathcal{I}} \mathcal{M}_\alpha := (\bigcup_{\alpha \in I} A_\alpha, <^{\mathcal{I},\mathfrak{S}}, \bigcup\nolimits_{\alpha \in I} {P_1}^\alpha, \dots, \bigcup_{\alpha \in I} {P_l}^\alpha),$$

where:

If $\alpha, \beta \in I$, $a \in A_\alpha$, $b \in A_\beta$, then $b <^{\mathcal{I},\mathfrak{S}} a$ iff $\beta <^{\mathcal{I}} \alpha$ or $\beta = \alpha$ and $b <^\alpha a$.

If the domains of the $\mathcal{M}_\alpha$'s are not disjoint, replace them with isomorphic chains that have disjoint domains, and proceed as before.

If $\mathcal{I} = (\{0, 1\}, <)$ and $\mathfrak{S} = (\mathcal{M}_0, \mathcal{M}_1)$, we denote $\sum_{\alpha \in \mathcal{I}} \mathcal{M}_\alpha$ by $\mathcal{M}_0 + \mathcal{M}_1$.

If $\mathcal{M}_\alpha$ is isomorphic to $\mathcal{M}$ for every $\alpha \in I$, we denote $\sum_{\alpha \in \mathcal{I}} \mathcal{M}_\alpha$ by $\mathcal{M} \times \mathcal{I}$.

The next proposition states that taking ordered sums preserves $\equiv^n$-equivalence.

Lemma 4.4 *Let $n \in \mathbb{N}$. Assume:*

1. *$(I, <^{\mathcal{I}})$ is a linear order,*
2. *$\big(\mathcal{M}^0_\alpha \mid \alpha \in I\big)$ and $\big(\mathcal{M}^1_\alpha \mid \alpha \in I\big)$ are sequences of chains (in the same signature), and*
3. *for every $\alpha \in I$, $\mathcal{M}^0_\alpha \equiv^n \mathcal{M}^1_\alpha$.*

Then, $\sum_{\alpha \in I} \mathcal{M}^0_\alpha \equiv^n \sum_{\alpha \in I} \mathcal{M}^1_\alpha$.

This allows us to define the sum of formulas in $Hin^n(<, P_1, \dots P_l)$ with respect to any linear order.

In particular, this theorem justifies the notation $\tau_0 + \tau_1$ for the n-type of a chain which is the ordered sum of two chains of n-types τ_0 and τ_1, respectively. Similarly, we write $\tau \times \omega$ for the n-type of a sum $\Sigma_{i \in \omega} \mathcal{M}_i$ where all $\mathcal{M}_i$ are of n-type τ; the n-type $\tau \times \omega^{-1}$ is defined similarly, where ω^{-1} is the order type of negative integers.

Another important operation on chains and on n-types is **shuffle**.

Let $\mathfrak{S} := (\mathcal{M}_\alpha \mid \alpha \in \mathbb{Q})$ be a sequence of chains indexed by the rationals. Let $Q_1, \dots, Q_k \subseteq \mathbb{Q}$ be a partition of $\mathbb{Q}$ into k everywhere dense sets. Let $\mathcal{N}_1, \dots, \mathcal{N}_k$ be chains. If for $i = 1, \dots, k$ and $q \in Q_i$, $\mathcal{M}_q$ is isomorphic to $\mathcal{N}_i$, we denote $\sum_{\alpha \in \mathbb{Q}} \mathcal{M}_\alpha$ by *shuffle*$(\mathcal{N}_1, \dots, \mathcal{N}_k)$. Note that different partitions of $\mathbb{Q}$ into k everywhere dense sets are isomorphic; hence, the shuffle is well defined. The corresponding operation on n-types will be also denoted by *shuffle*.

5 Recursively Defined Classes of Structures

Let Δ be a signature and $k \in \mathbb{N}$. A k-ary Δ-*operator* is a function F which assigns to every k-tuple of Δ-structures a Δ structure. A *finite-set* Δ-*operator* is a function F which assigns to every finite set of Δ-structures a Δ structure. A Δ-*operator* is a k-ary ($k \in \mathbb{N}$) or a finite-set Δ-operator.

Let $\mathcal{C}$ be a set of Δ-structures. $\mathcal{C}$ is closed under a Δ-operator F if the result of application of F to structures from $\mathcal{C}$ is in $\mathcal{C}$.

Let $\mathcal{C}$ be a set of Δ-structures and $\mathfrak{F}$ be a family of Δ-operators. The *closure* of $\mathcal{C}$ under $\mathfrak{F}$ is the minimal class $\mathcal{C}'$ of Δ-structure which contains $\mathcal{C}$ and is closed under $\mathfrak{F}$. We denote this class by $Cl(\mathcal{C}, \mathfrak{F})$. It is said to be *recursively defined* from $\mathcal{C}$ by $\mathfrak{F}$.

Let $Cl^0(\mathcal{C}, \mathfrak{F}) := C$ and for $i \in \mathbb{N}$ define $Cl^{i+1}(\mathcal{C}, \mathfrak{F}) := Cl^i(\mathcal{C}, \mathfrak{F}) \cup \{\mathcal{M} \mid \mathcal{M} = F(\mathcal{M}_1, \dots, \mathcal{M}_k)$ for k-ary $F \in \mathfrak{F}$ and $\mathcal{M}_j \in Cl^i(\mathcal{C}, \mathfrak{F})\} \cup \{\mathcal{M} \mid \mathcal{M} = F(\mathcal{A})$ for finite-set operator $F \in \mathfrak{F}$ and $\mathcal{A} \subseteq Cl^i(\mathcal{C}, \mathfrak{F})\}$. Define $Cl^*(\mathcal{C}, \mathfrak{F}) := \cup_{i \in \mathbb{N}} Cl^i(\mathcal{C}, \mathfrak{F})$. Note that $Cl^*(\mathcal{C}, \mathfrak{F}) = Cl(\mathcal{C}, \mathfrak{F})$.

Let $\sim$ be an equivalence on Δ-structures. The *index* of $\sim$ is the cardinality of the set of $\sim$-equivalence classes; $\sim$ has a *finite index* if there are only finitely many $\sim$-equivalence classes.

A k-ary Δ operator F *respects* $\sim$ if for Δ-structures $\mathcal{M}_1, \dots, \mathcal{M}_k, \mathcal{N}_1, \dots, \mathcal{N}_k$

$$F(\mathcal{M}_1, \dots, \mathcal{M}_k) \sim F(\mathcal{N}_1, \dots, \mathcal{N}_k)$$

whenever $\mathcal{M}_i \sim \mathcal{N}_i$ $(i = 1, \dots, k)$.

If F respects $\sim$, then it induces a k-ary operation on the $\sim$-equivalence classes. We denote this operation by F as it will always be clear from the context whether we use an operator on Δ-structures or the corresponding operation on the $\sim$-equivalence classes.

If $\mathcal{A}$ and $\mathcal{B}$ are sets of Δ-structures, we say that $\mathcal{A}$ is $\sim$-equivalent to $\mathcal{B}$ if $\forall \mathcal{M} \in \mathcal{A} \exists \mathcal{N} \in \mathcal{B} (\mathcal{M} \sim \mathcal{N})$ and $\forall \mathcal{M} \in \mathcal{B} \exists \mathcal{N} \in \mathcal{A} (\mathcal{M} \sim \mathcal{N})$.

A finite-set Δ-operator *respects* $\sim$ if $F(\mathcal{A}) \sim F(\mathcal{B})$ whenever $\mathcal{A} \sim \mathcal{B}$.

If a finite-set operator F respects $\sim$, then it induces an operation which assigns a $\sim$-equivalence class to every finite subset of $\sim$-equivalence classes.

A family $\mathfrak{F}$ of Δ-operators *respects* $\sim$ if every operator in $\mathfrak{F}$ respects $\sim$.

Lemma 5.1 *Assume that $\sim$ is an equivalence of finite index l, and $\mathfrak{F}$ respects $\sim$. Then for every $\mathcal{M} \in Cl(\mathcal{C}, \mathfrak{F})$ there is $\mathcal{N} \in Cl^l(\mathcal{C}, \mathfrak{F})$ such that $\mathcal{M} \sim \mathcal{N}$.*

Proof. Let E_n be the set of $\sim$-equivalence classes of structures from $Cl^n(\mathcal{C}, \mathfrak{F})$. Then, $\forall n E_n \subseteq E_{n+1}$. Hence, there is $i \leq l$ such that $E_i = E_{i+1}$. This implies that $\forall j E_i = E_{i+j}$. In particular, $\forall j E_l \supseteq E_j$, therefore, the lemma holds. □

For every n the set of operators $\{+, \times\omega, \times\omega^{-1}, \textit{shuffle}\}$ respects $\equiv^n$.

Strictly speaking, these are polymorphic operators. For every set $\overline{P}$ of monadic predicate names, there is a corresponding binary operator $+$ on $\overline{P}$-labeled chains.

Recall that for a Δ-structure $\mathcal{M}$ and $\Delta' \subseteq \Delta$ the Δ' reduct of $\mathcal{M}$ on Δ' is a Δ'-structure which has the same domain as $\mathcal{M}$ and the same interpretation of symbols from Δ'. We denote by $\mathcal{M}|\Delta'$ the reduct of $\mathcal{M}$ on Δ'.

The reduct distributes over the sum in the following sense:

The reduct distributes over +

> Let $\overline{P'} \subseteq \overline{P}$ be sets of monadic predicate names, let $\mathcal{M}$ and $\mathcal{N}$ be $\overline{P}$-chains. Then $(\mathcal{M}+\mathcal{N})|\{<, \overline{P'}\}$ and $(\mathcal{M}|\{<, \overline{P'}\}) + (\mathcal{N}|\{<, \overline{P'}\})$ are isomorphic.

The reduct also distributes over $\{\times\omega, \times\omega^{-1}, \mathit{shuffle}\}$.

Let $\overline{P}$ be a set of monadic predicate names, let $\overline{P}_1, \dots, \overline{P}_k \subseteq \overline{P}$ be a sequence of subsets of $\overline{P}$, and let $\mathcal{M}$ be a $\overline{P}$-chain. Define $\mathit{ptype}^n(\mathcal{M}; (\overline{P}_1, \dots, \overline{P}_k))$, the *product n-type* of $\mathcal{M}$ with respect to $\overline{P}_1, \dots, \overline{P}_k$, as

$$\mathit{ptype}^n(\mathcal{M}; (\overline{P}_1, \dots, \overline{P}_k)) := (\tau_1, \dots, \tau_k)\,,$$

where $\tau_i = \mathit{type}^n(\mathcal{M}|\{<, \overline{P}_i\})$ be the n-types of the reduct.

For a class $\mathcal{C}$ of $\overline{P}$-chain,

$$\mathit{ptype}^n(\mathcal{C}; (\overline{P}_1, \dots, \overline{P}_k)) := \{\mathit{ptype}^n(\mathcal{M}; (\overline{P}_1, \dots, \overline{P}_k)) \mid \mathcal{M} \in \mathcal{C}\}.$$

Lemma 5.2 *1. If $\mathit{ptype}^n(\mathcal{M}^i; (\overline{P}_1, \dots, \overline{P}_k)) = (\tau_1^i, \dots, \tau_k^i)$ for $i \in \{0,1\}$, then*

$$\mathit{ptype}^n(\mathcal{M}^0 + \mathcal{M}^1; (\overline{P}_1, \dots, \overline{P}_k)) = (\tau_1^0 + \tau_1^1, \dots, \tau_k^0 + \tau_k^1)$$

2. If $\mathit{ptype}^n(\mathcal{M}; (\overline{P}_1, \dots, \overline{P}_k)) = (\tau_1, \dots, \tau_k)$, then

$$\mathit{ptype}^n(\mathcal{M} \times \omega; (\overline{P}_1, \dots, \overline{P}_k)) = (\tau_1 \times \omega, \dots, \tau_k \times \omega)$$

$$\mathit{ptype}^n(\mathcal{M} \times \omega^{-1}; (\overline{P}_1, \dots, \overline{P}_k)) = (\tau_1 \times \omega^{-1}, \dots, \tau_k \times \omega^{-1})$$

3. if $\mathcal{A}$ is a finite set of structures and for $j = 1, \dots k$, and $U_j = \{\tau_j \mid \mathit{ptype}^n(\mathcal{M}; (\overline{P}_1, \dots, \overline{P}_k)) = (\tau_1, \dots, \tau_j, \dots, \tau_k) \wedge \mathcal{M} \in \mathcal{A}\}$, then $\mathit{ptype}^n(\mathit{shuffle}(\mathcal{A}); (\overline{P}_1, \dots, \overline{P}_k)) = (\mathit{shuffle}(U_1), \dots, \mathit{shuffle}(U_k))$.

6 EXPTIME Algorithm

In this section we present an EXPTIME algorithm for the satisfiability of conjunctive formulas.

Let Φ be a finite set of formulas of quantifier depth $\le n$ in the first-order monadic logic over $\{<\}$ with free variables among $X_1, \dots, X_m$.

Let $\psi = \varphi_1(\overline{P_1}) \wedge \dots \wedge \varphi_k(\overline{P_k})$ be a Φ-conjunctive formula. Let $\mathfrak{F} := \{+, \times\omega, \times\omega^{-1}, \mathit{shuffle}\}$. Let $\mathcal{C}$ be a set of structures over signature $\{<, \cup_{i=1}^k \overline{P_i}\}$. Recall that $\mathfrak{F}$ respects $\equiv^n$, therefore, by Lemma 5.1, ψ is satisfiable over $\mathit{Cl}(\mathcal{C}, \mathfrak{F})$ if it is satisfiable over $\mathit{Cl}^l(\mathcal{C}, \mathfrak{F})$, where $l := |\mathit{Hin}^n(<, \cup_{i=1}^k \overline{P_i})|$ is the cardinality of the set $\mathit{Hin}^n(<, \cup_{i=1}^k \overline{P_i})$ of Hintikka formulas. This l grows like the n-time iterated exponential function $\exp(n,k)$ ($\exp(1,x) := 2^x$ and $\exp(i+1,x) := 2^{\exp(i,x)}$). We replace this bound by a bound exponential in k and derive an exponential time algorithm for the satisfiability of Φ-conjunctive formulas over $\mathit{Cl}(\mathcal{C}, \mathfrak{F})$. Our arguments are valid not only for this recursively defined class, but for any recursive class which is definable by a finite set of operators that respect $\equiv^n$-equivalence and satisfy an analog of Lemma 5.2.

Lemma 6.1 *Let Φ be a finite set of formulas of the quantifier depth $\leq n$ in the first-order monadic logic over $\{<\}$ with free variables among $X_1, \dots, X_m$. A Φ-conjunctive formula $\varphi_1(\overline{P_1}) \wedge \dots \wedge \varphi_k(\overline{P_k})$ is satisfiable in $\mathcal{M}$ if and only if $ptype^n(\mathcal{M}; (\overline{P}_1, \dots, \overline{P}_k)) = (\tau_1, \dots, \tau_k)$ and $\tau_i(\overline{P_i}) \rightarrow \varphi_i(\overline{P_i})$ for $i = 1, \dots, k$.*

Define the equivalence $\sim^n_{(\overline{P}_1,\dots,\overline{P}_k)}$ on chains over the signature $\{<, \cup_{i=1}^k \overline{P_i}\}$ as $\mathcal{M} \sim^n_{(\overline{P}_1,\dots,\overline{P}_k)} \mathcal{N}$ iff $ptype^n(\mathcal{M}; (\overline{P}_1, \dots, \overline{P}_k)) = ptype^n(\mathcal{N}; (\overline{P}_1, \dots, \overline{P}_k))$. The number of $\sim^n_{(\overline{P}_1,\dots,\overline{P}_k)}$ equivalence classes is $\leq |Hin^n(<, P_1, \dots, P_m)|^k$; hence, it is at most exponential in k. $\mathfrak{F}$ respects $\sim^n_{(\overline{P}_1,\dots,\overline{P}_k)}$. Therefore, by Lemma 5.1, we obtain:

Lemma 6.2 *For every finite set Φ of first-order formulas there is c_Φ such that a Φ-conjunctive formula $\psi = \varphi_1(\overline{P_1}) \wedge \dots \wedge \varphi_k(\overline{P_k})$ is satisfiable in $Cl(\mathcal{C}, \mathfrak{F})$ iff it is satisfiable in $Cl^{{c_\Phi}^k}(\mathcal{C}, \mathfrak{F})$.*

Consider the following problem.

Membership Problem for fixed $n, m \in \mathbb{N}$; all tuples $\overline{P}_i$ are of length $\leq m$.

Input: $\overline{\tau} = (\tau_1 \dots \tau_k) \in Hin^n(<, \overline{P}_1) \times \dots \times Hin^n(<, \overline{P}_k)$ and an oracle I for membership in $ptype^n(\mathcal{C}; (\overline{P}_1, \dots, \overline{P}_k))$.
Question: Is $\overline{\tau}$ in $ptype^n(Cl(\mathcal{C}, \mathfrak{F}); (\overline{P}_1, \dots, \overline{P}_k))$?

Lemma 6.3 *The membership problem is in $EXPTIME^I$.*

Proof. Our algorithm is presented below.

Algorithm 1 Membership Problem is in EXPTIME^I

$R \leftarrow I$ { i.e., for every $\overline{\tau}$ if $\overline{\tau} \in I$ then add $\overline{\tau}$ to R.}
Updated $\leftarrow$ True.
while Updated **do**
1. Updated $\leftarrow$ False;
2. Compute $R' = Cl^1((R, +)$; **If** $R' \neq R$ **then** Updated $\leftarrow$ True;
3. $R \leftarrow R'$; Compute $R' = Cl^1(R, \times\omega)$; **If** $R' \neq R$ **then** Updated $\leftarrow$ True;
4. $R \leftarrow R'$; Compute $R' = Cl^1(R, \times\omega^{-1})$; **If** $R' \neq R$ **then** Updated $\leftarrow$ True;
5. $R \leftarrow R'$; Compute $R' = Cl^1(R, \textit{shuffle})$; **If** $R' \neq R$ **then** Updated $\leftarrow$ True;

end while
if $\overline{\tau} \in R$ **return** True.

Let $N_0 = |Hin^n(<, X_1, \dots, X_m)|$. The number of iterations of the loop is bounded by N_0^k.

$R' = Cl^1((R, +)$ can be computed in time $O(N_0^{2k})$ as follows. Let $R' \leftarrow R$. For each pair $\overline{\tau} = (\tau_1, \dots, \tau_k), \overline{\tau'} = (\tau'_1, \dots, \tau'_k) \in R$ add $(\tau_1 + \tau'_1, \dots, \tau_k + \tau'_k)$ to R'. Hence, Step 2 can be implemented in time $O(N_0^{2k})$.

Steps 3 and 4 can be implemented in $O(N_0^k)$.

The computation of $R' = Cl^1(R, \mathit{shuffle})$ is more subtle. Indeed, a naive approach can try to compute *shuffle* for every subset of R. However, the number of such subsets is $2^{{N_0}^k}$ and it is double-exponential. $R' = Cl^1(R, \mathit{shuffle})$ can be computed in EXPTIME as follows:

Algorithm 2 Computation of $Cl^1(R, \mathit{shuffle})$

Let $H_i := \mathcal{P}(Hin^n(<, \overline{P_i}))$ be the set of subsets of $Hin^n(<, \overline{P_i})$.
for every $U = (U_1, \ldots, U_k) \in H_1 \times \cdots \times H_k$ **do**
{ *Check if there is a sequence* $(\tau_1^1, \ldots, \tau_k^1), \ldots, (\tau_1^m, \ldots, \tau_k^m) \in R$ *such that* $U_i = \{\tau_i^j \mid j \leq m\}$ *and update* R' *as follows:* }

1. $(B_1, \ldots, B_k) \leftarrow (U_1, \ldots, U_k)$;
2. **for** every $\overline{\tau} = (\tau_1, \ldots \tau_k) \in R$ **if** $\wedge_i \tau_i \in U_i$ **then** $B_i \leftarrow B_i \setminus \{\tau_i\}$;
3. **If** $\wedge B_i = \emptyset$ **then** {such a sequence exists, and we have to update R'}
   $R' \leftarrow R' \cup \{(\mathit{shuffle}(U_1), \ldots, \mathit{shuffle}(U_k))\}$;

end for

The number of iterations of the external loop is $2^{N_0 k}$ and the number of iterations of the internal loop is bounded by N_0^k. Hence, Step 5 can be implemented in time $O(2^{N_0 k} \times N_0^k)$.

Since every step can be implemented in EXPTIME and the number of iterations is exponential, we obtain that the membership problem is in EXPTIME with the oracle I. □

Let *One* be the class of one-element chains. It is clear that we can decide in EXPTIME, whether $\tau \in ptype^n(One; (\overline{P}_1, \ldots, \overline{P}_k))$. Hence, as a consequence of Lemma 6.3, we obtain:

Proposition 6.4 *The satisfiability problem for Φ-conjunctive formulas over the class $Cl(One, \mathfrak{F})$ is in EXPTIME.*

Proof. For every $\varphi \in \Phi$ we can pre-compute the set $H_\varphi := \{\tau \in Hin^n(<, X_1, \ldots, X_m) \mid \tau \rightarrow \varphi\}$ (this depends only on Φ and is independent from the input).

Let $\psi = \varphi_1(\overline{P_1}) \wedge \cdots \wedge \varphi_k(\overline{P_k})$ be a Φ-conjunctive formula. First compute the set S of all $\overline{\tau}$ in $ptype^n(Cl(One, \mathfrak{F}); (\overline{P}_1, \ldots, \overline{P}_k))$. The cardinality of S is at most exponential. By the previous lemma, S can be computed in EXPTIME. Then, by Lemma 6.1, it is enough to check whether there is $(\tau_1, \ldots, \tau_k) \in S$ such that $\tau_i(\overline{P_i}) \rightarrow \varphi_i(\overline{P_i})$ for $i = 1, \ldots, k$. This can be done in EXPTIME using the pre-computed sets H_φ. □

Läuchli and Leonard [13] proved[1] the following theorem:

[1] Läuchli and Leonard considered the logic with the order relation only. Their proof can be adapted easily to the first-order monadic logic over chains.

Theorem 6.5 *A first-order formula is satisfiable over a linear order if it is satisfiable over* $Cl(One, \mathfrak{F})$.

As a consequence of Theorem 6.5 and Propositions 6.4 and 3.1 we obtain:

Theorem 6.6 *Let TL be a temporal logic with a finite set of* $\exists$*-MSO definable modalities. The satisfiability problem for TL over the class of chains is in* EXPTIME.

In the next section we will show that EXPTIME upper bound can be replaced by PSPACE upper bound.

Let us conclude this section by a remark on optimality of our algorithm. The only properties of operators $\{+, \times\omega, \times\omega^{-1}, \mathit{shuffle}\}$ which were used in our EXPTIME algorithm are (1) they respect $\equiv^n$ and (2) the reduct distributes over these operators. If $\mathfrak{F}$ is any set of operators with these properties, then the membership problem for $Cl(One, \mathfrak{F})$ is in EXPTIME.

Below we will show that for such $\mathfrak{F}$ in general EXPTIME bound cannot be improved.

Let $\Delta_2 = \{<, \mathit{Left}, \mathit{Right}\}$ be a signature, where $<$ is a binary predicate and $\mathit{Left}, \mathit{Right}$ are unary predicates. We will interpret Δ_2 over the binary trees, where $<$ is the ancestor relation and *Left* (respectively, *Right*) are interpreted as the set of left (respectively, right) children. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be binary trees expanded by unary predicates $P_1, \dots, P_k$, and let R be a one element chain for these predicate names. We assume that the domains of $\mathcal{M}_1, \mathcal{M}_2$ and R are disjoint and define a ternary operation $\boxplus(\mathcal{M}_1, R, \mathcal{M}_2)$ as follows. $\boxplus(\mathcal{M}_1, R, \mathcal{M}_2)$ is a binary tree; its domain is the union of the domains of $\mathcal{M}_1$, R and $\mathcal{M}_2$; the unique node r of R is the root of this tree. The left and right subtrees of r are $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively. Predicate name P_i is interpreted as the union of its interpretations in $\mathcal{M}_1, R$ and $\mathcal{M}_2$.

The operation $\boxplus$ has properties (1) and (2). The closure of *One* under $\boxplus$ is the set of all finite binary trees. As a consequence, we can derive that the satisfiability problem for any temporal logics with a finite set of $\exists$-MSO definable modalities over the class of finite binary trees is in EXPTIME. Note that CTL can be described as a temporal logic with a finite set of modalities definable in $\exists$-MSO and the satisfiability problem for CTL over the class of finite binary trees is EXPTIME hard. Hence, in general our EXPTIME upper bound for the satisfiability problem over recursively definable classes is optimal.

7 PSPACE Algorithm

Let $\mathfrak{F} = \{+, \times\omega, \times\omega^{-1}, \mathit{shuffle}\}$. To every chain in $Cl(One, \mathfrak{F})$ we assign a natural number - the rank of a chain. Define sets $\mathcal{C}^{\leq i} \subseteq Cl(One, \mathfrak{F})$ as follows:

1. $\mathcal{C}^{\leq 0}$ is the set of finite chains.
2. $\mathcal{C}^{\leq i+1}$ is the closure under $+$ of the union of $\mathcal{C}^{\leq i}$, $\{\mathcal{M} \times \omega \mid \mathcal{M} \in \mathcal{C}^{\leq i}\}$, $\{\mathcal{M} \times \omega^{-1} \mid \mathcal{M} \in \mathcal{C}^{\leq i}\}$ and $\{\mathit{shuffle}(\mathcal{A}) \mid \mathcal{A}$ is a finite subset of $\mathcal{C}^{\leq i}\}$.

A chain $\mathcal{M}$ has *rank* $i+1$ if $\mathcal{M} \in \mathcal{C}^{\leq i+1} \wedge \mathcal{M} \notin \mathcal{C}^{\leq i}$.

Every chain of a finite rank can be described by its finite construction tree. Let $\overline{P}$ be a set of monadic predicate names. A construction tree T for $\overline{P}$-chains is a labeled tree which has the following properties: the leaves of T are labeled by one-element $\overline{P}$-chains; the internal nodes are labeled by $+, \times\omega, \times\omega^{-1}$ and *shuffle*; a node labeled by $\times\omega$ or by $\times\omega^{-1}$ has one child; a node labeled by $+$ has at least two children and these children are linearly ordered; a node labeled by *shuffle* has at least one child.

Let T be a construction tree. A chain $[|T|]$, assigned to T, is defined as follows:

1. if T is a one-element tree then $[|T|]$ is the one-element chain which is the label of its only node.
2. If the root of T is labeled by $\times\omega$ (or by $\times\omega^{-1}$), then $[|T|]$ is $[|T_1|] \times \omega$ (respectively, $[|T_1|] \times \omega^{-1}$)) where T_1 is the subtree of T rooted at the child of its root.
3. If the root of T is labeled by $+$ and its children (ordered from younger to older) are trees $T_1, \dots, T_m$ then $[|T|] := [|T_1|] + \dots + [|T_m|]$.
4. If the root of T is labeled by *shuffle* and its children are trees $T_1, \dots, T_m$ then $[|T|] := \textit{shuffle}([|T_1|], \dots, [|T_m|])$.

Lemma 7.1 *If a chain $\mathcal{M}$ has rank $\leq i$, then there is a chain construction tree T such that $\mathcal{M} = [|T|]$ and the height of T is bounded by $2i+1$.*

Proof. A chain $\mathcal{M}$ has rank $\leq i$ if there is a tree T such that $\mathcal{M} = [|T|]$ and the number of nodes labeled by $\times\omega$, $\times\omega^{-1}$ and *shuffle* on any path from the root to a leaf is bounded by i (we do not count nodes labeled by $+$). For every tree T there is a tree T' such that $[|T'|] = [|T|]$ and no $+$ node has a child labeled by $+$. Indeed, if a $+$ node v of T has as a child a $+$ node u we can remove u and make its children to be children of v (between the left and the right brothers of u). Hence, if a chain $\mathcal{M}$ has rank $\leq i$ then there is a tree T such that $\mathcal{M} = [|T|]$ and the height of T is bounded by $2i+1$. ☐

We are going to present a PSPACE algorithm for the satisfiability problem for Φ-conjunctive formulas. Its correctness and complexity analysis are based on the following Lemma which refines Lemma 6.2 and will be proved in Sect. 9.

Lemma 7.2 (small rank property) *For every finite set Φ of first-order formulas there is r_Φ such that every Φ-conjunctive formula $\psi = \varphi_1(\overline{P_1}) \wedge \dots \wedge \varphi_k(\overline{P_k})$ is satisfiable in $Cl(One, \mathfrak{F})$ iff it is satisfiable in a chain of rank $\leq k \times r_\Phi$.*

By Theorem 6.5, Lemma 7.1, and Lemma 7.2, $\varphi_1(\overline{P_1}) \wedge \dots \wedge \varphi_k(\overline{P_k})$ is satisfiable iff

- **(A)** there is a chain construction tree T of height $\leq 2k \times r_\Phi + 1$ such that $\textit{ptype}^n([|T|]; (\overline{P}_1, \dots, \overline{P}_k)) = (\tau_1, \dots, \tau_k)$ and
- **(B)** $\tau_i \rightarrow \varphi_i$ for $i = 1, \dots, k$.

Now, we are ready to improve our EXPTIME bound of Theorem 6.6 to PSPACE.

Theorem 7.3 *Let TL be a temporal logic with a finite set of ∃-MSO definable modalities. The satisfiability problem for TL over the class of chains is in* PSPACE.

By proposition 3.1 it is sufficient to provide a PSPACE algorithm for the satisfiability of Φ-conjunctive formulas. Let $\psi = \varphi_1(\overline{P_1}) \wedge \cdots \wedge \varphi_k(\overline{P_k})$ be such a formula. Our algorithm guesses $(\tau_1, \ldots, \tau_k)$ and checks in linear time condition (B). Then the non-deterministic algorithm SAT, defined below, checks (A). SAT works in polynomial space in k, assuming that the last argument is polynomial in k which is the case with $N = 2k \times r_\Phi + 1$. Fig. 1 contains the definition of the algorithm SAT (some details are omitted).

Membership Problem

Input 1. $(\tau_1, \ldots, \tau_k)$, where $\tau_i \in Hin^n(<, \overline{P}_i)$ and $\overline{P}_i \subseteq \overline{P}$ are sets of l predicate names (note that n and l are fixed and are not part of the input).
2. $N \in \mathbb{N}$.

Output True, if there is a construction tree T of height $\leq N$ such that $ptype^n([|T|]; (\overline{P}_1, \ldots, \overline{P}_k)) = (\tau_1, \ldots, \tau_k)$.

- If $N = 0$ and there is a one element chain $\mathcal{M}$ such that $ptype^n(\mathcal{M}; (\overline{P}_1, \ldots, \overline{P}_k)) = (\tau_1, \ldots, \tau_k)$ then return True;
- Go non-deterministically to 1-5.
  - **(1.)** Return SAT$((\tau_1, \ldots, \tau_k), N-1)$.
  - **(2.)** Guess $(\tau'_1, \ldots, \tau'_k)$ such that SAT$((\tau'_1, \ldots, \tau'_k), N-1)$ returns True and $\tau_i = \tau'_i \times \omega$ for $0 < i \leq k$.
  - **(3.)** Guess $(\tau'_1, \ldots, \tau'_k)$ such that SAT$((\tau'_1, \ldots, \tau'_k), N-1)$ returns True and $\tau_i = \tau'_i \times \omega^{-1}$ for $0 < i \leq k$.
  - **(4.)** Guess on-the-fly a sequence

$$(\tau_1^1, \ldots, \tau_k^1), (\tau_1^2, \ldots, \tau_k^2), \ldots, (\tau_1^m, \ldots, \tau_k^m)$$

    such that
    (4.1) for $0 < i \leq m$, SAT$((\tau_1^i, \ldots, \tau_k^i), N-1)$ returns True,
    (4.2) for $0 < j \leq k$, $\tau_j = \tau_j^1 + \ldots + \tau_j^m$.
  - **(5.)** Guess $(U_1, \ldots, U_k)$, where $U_i \subseteq Hin^n(<, \overline{P_i})$ such that
    (5.1) for $0 < j \leq k$, $\tau_j = shuffle(U_j)$
    and guess on-the-fly a sequence

$$(\tau_1^1, \ldots, \tau_k^1), (\tau_1^2, \ldots, \tau_k^2), \ldots, (\tau_1^m, \ldots, \tau_k^m)$$

    such that
    (5.2) for $0 < i \leq m$, SAT$((\tau_1^i, \ldots, \tau_k^i), N-1)$ returns True,
    (5.3) for $0 < j \leq k$, $U_j = \{\tau_j^i \mid i \leq m\}$.

Figure 1: Algorithm SAT

Since $+$ is associative, to verify condition (4.2) we need to keep in the memory at every stage p only two tuples: the tuple of the partial sum $\left(\sum_{s=1}^{s<p} \tau_1^s, \ldots, \sum_{s=1}^{s<p} \tau_k^s\right)$ and the current guess $(\tau_1^p, \ldots, \tau_k^p)$. The tuple of the partial sums can be easily updated. We can assume that all partial sums are different; hence, m is bounded by the number of possible $ptype^n(\mathcal{M}; (\overline{P}_1, \ldots, \overline{P}_k))$ which is bounded by $|Hin^n(< .X_1, \ldots, X_l)|^k$ and the counter for m can be saved in space linear in k.

To verify condition (5.3) we need to keep in memory at every stage p only two tuples: the tuple $U_i^p = \{\tau_i^s \mid s < p\}$ (for $i = 1, \ldots, k$) and the current guess $(\tau_1^p, \ldots, \tau_k^p)$. We have to verify that $(\tau_1^p, \ldots, \tau_k^p)$ is in $(U_1, \ldots, U_k)$, i.e., $\tau_i^p \in U_i$ and update the tuple $(U_1^p, \ldots, U_k^p)$. In (5.) we can assume that no tuple occurs twice; hence, m is bounded by the number of possible $ptype^n(\mathcal{M}; (\overline{P}_1, \ldots, \overline{P}_k))$ and the counter for m can be saved in space linear in k.

The depth of recursion is bounded by N. Hence, SAT works in non-deterministic space $O(kN)$.

In order to check (A) we call SAT with $N = 2r_{\Phi} \times k + 1$. Therefore, our procedure works in non-deterministic polynomial space and by the Savitch theorem it can be implemented by a deterministic PSPACE algorithm.

The next two sections are geared towards the proof of Lemma 7.2.

8 Automata on Linear Orders

Büchi used finite automata over ω-words to prove that monadic second-order logic is decidable over ω. In order to prove the decidability of monadic second-order logic over countable ordinals, Büchi introduced finite automata on words of ordinal length [4]. Büchi's model extends traditional finite automata using limit transitions to handle positions with no predecessor. He proved that over countable ordinals these automata are equivalent to monadic second-order logic.

These automata were extended to finite automata on linear orderings by Bruyère and Carton [2]. This model further extends traditional finite automata using limit transitions to handle positions with no successor or no predecessor. In [18] it was shown that these automata can be complemented over countable scattered linear orderings and are equivalent to monadic second-order logic over the countable scattered linear orderings. However, this equivalence fails over dense orders and over uncountable orders [1].

We first recall some basic definitions about linear orders. Then, we introduce finite base automata which have the same expressive power as finite state automata of [2], but are more appropriate for our purposes.

In order to define the runs of an automaton, we use the notion of cut. A *cut* of a linear order J is a partition (L, U) of J such that $a < b$ for any $a \in L$ and $b \in U$. A cut (L, U) is a *gap* if neither L has a maximal element, nor U has a minimal element and $L \neq \emptyset \neq U$. An order is *Dedekind-complete* if it does not have gaps. We denote by $\widehat{J}$ the set of cuts of J. This set is equipped with the order defined by $(L_1, U_1) < (L_2, U_2)$ if $L_1 \subsetneq L_2$. This ordering on $\widehat{J}$ can be extended to $J \cup \widehat{J}$. in a natural way: $(L, U) < a$ if $a \in U$. The order $\widehat{J}$ is Dedekind-complete. Its minimal (maximal) element is $\widehat{J}_{\min} = (\emptyset, J)$ (respectively, $\widehat{J}_{\max} = (J, \emptyset)$). For any element a of J, there are two successive cuts: $a^- := (\{b \in J \mid b < a\}, \{b \in J \mid b \geq a\})$ and $a^+ := (\{b \in J \mid b \leq a\}, \{b \in J \mid b > a\})$. Note that if If $a < b$ are consecutive elements of J then a^+ and b^- denote the same cut.

Given an alphabet Σ, a Σ-word of length J is a sequence $(\sigma_a \mid a \in J)$ of elements of Σ indexed by J.

In [7] we introduced simple ordinal automata which work over words of ordinal length. We extend this definition to finite base automata working on words over arbitrary linear orders.

Finite base automata have the same expressive power as finite state automata over chains. An important parameter of a finite base automaton is the size of its base. An advantage of finite base automata over finite state automata is that taking the conjunction is easy and the base of an automaton for the conjunction grows linearly in the number of conjuncts.

Definition 8.1 (finite base automata) *A finite base automaton $\mathfrak{A}$ is a tuple of the form $(B, Q, \Sigma, \delta_{next}, \delta_{\lim}, Q_{init}, Q_{fin})$ such that*

- *B is a finite set (the basis of $\mathfrak{A}$),*
- *$Q \subseteq \mathcal{P}(B)$ (the set of states),*
- *$Q_{init}, Q_{fin} \subseteq Q$ (the sets of initial states and final states),*
- *Σ is a finite alphabet,*
- *$\delta_{next} \subseteq Q \times \Sigma \times Q$ is the next-step transition relation,*
- *$\delta_{\lim} \subseteq \mathcal{P}(B) \times Q \cup Q \times \mathcal{P}(B)$ is the limit transition relation.*

Let f be a function from a set I into $\mathcal{P}(B)$. Define

$$\text{always}(f) := \{b \in B \mid \forall c \in I \quad b \in f(c)\}.$$

If I is a linear order, we define the left and right base-limit sets of f at $c \in I$ as the sets of base elements that appear in every state arbitrarily close to c (respectively, to its left and to its right). Formally, $Base_{\underrightarrow{\lim}}(c, f)$ is defined as

$$Base_{\underrightarrow{\lim}}(c, f) := \{b \in B \mid \forall a < c \exists d (a < d < c) \wedge b \in \text{always}(f \lfloor (d, c)\},$$

where $f \lfloor (d, c)$ is the restriction of f to the interval (d, c).

$Base_{\underleftarrow{\lim}}(c, f)$ is defined similarly.

Given a finite base automaton $\mathfrak{A}$, a *run* of $\mathfrak{A}$ on Σ-word s over a linear order $\mathcal{I}$ is a function $\rho : \widehat{\mathcal{I}} \to Q$ such that

- For each $c \in \mathcal{I}$, $\rho(c^-) \xrightarrow{s(c)} \rho(c^+)$,
- if $c \in \widehat{\mathcal{I}} \setminus \widehat{\mathcal{I}}_{\min}$ has no predecessor, $(Base_{\underrightarrow{\lim}}(c, \rho), \rho(c)) \in \delta_{\lim}$, and
- if $c \in \widehat{\mathcal{I}} \setminus \widehat{\mathcal{I}}_{\max}$ has no successor, $(\rho(c), Base_{\underleftarrow{\lim}}(c, \rho)) \in \delta_{\lim}$.

An $\mathfrak{A}$-run ρ is *accepting* if $\rho(\widehat{\mathcal{I}}_{\min}) \in Q_{init}$ and $\rho(\widehat{\mathcal{I}}_{\max}) \in Q_{fin}$. $\mathfrak{A}$ *accepts* a word s if there is an accepting run on s.

Let $\mathfrak{A}_1, \ldots, \mathfrak{A}_m$ be finite base automata. One can easily construct an automaton $\mathfrak{A}$ that accepts the intersection of the languages accepted by these automata. The number of states in $\mathfrak{A}$ is the product of the numbers of states of $\mathfrak{A}_i$ and this grows exponentially in m; however, the base size of $\mathfrak{A}$ is the sum of the base sizes of $\mathfrak{A}_i$.

Lemma 8.2 (intersection of finite base automata) *Let $\mathfrak{A}_1$ and $\mathfrak{A}_2$ be finite base automata. Assume that the base size of $\mathfrak{A}_1$ and $\mathfrak{A}_2$ are n_1 and n_2. There is a finite base automaton $\mathfrak{A}$ such that the base size of $\mathfrak{A}$ is $n_1 + n_2$ and a word s is accepted by $\mathfrak{A}$ iff it is accepted by $\mathfrak{A}_1$ and by $\mathfrak{A}_2$.*

A word $s := (\sigma_a \mid a \in J)$ indexed by J over an alphabet $\{0,1\}^k$ can be identified with a chain $(J, <, P_1, \ldots, P_k)$ over J where $P_i = \{a \in J \mid \text{ the } i\text{-th bit of } \sigma_a = 1\}$. This is a bijection between the $\{0,1\}^k$-words over J and the chain with k monadic predicates over J.

An automaton is said to be equivalent to a formula $\varphi(P_1, \ldots, P_k)$ over a class $\mathcal{C}$ of linear orders if for every linear order $J \in \mathcal{C}$ and every word s indexed by J, $\mathfrak{A}$ accepts s if and only if the corresponding chain satisfies φ.

Cristau [6] proved that every formula of the first-order fragment of the monadic logic is equivalent (over the class of all linear orders) to a finite-state automaton. Hence,

Theorem 8.3 *For every first-order formula φ there is a finite base automaton $\mathfrak{A}_\varphi$ equivalent to φ over the class of all linear orders.*

9 Small Rank Property

Let $\mathfrak{A}$ be a finite base automaton, $\mathcal{L}$ a chain and $\rho : \widehat{\mathcal{L}} \to Q$ be a run of $\mathfrak{A}$ on $\mathcal{L}$. $type_{\mathfrak{A}}(\rho) := (q, D, q')$, where $\rho(\widehat{\mathcal{L}}_{\min}) = q$, $\rho(\widehat{\mathcal{L}}_{\max}) = q'$ and $D := \text{always}(\rho)$.

If $type_{\mathfrak{A}}(\rho) := (q, D, q')$ we sometimes write $\rho : q \xrightarrow{D} q'$; we write $\rho :\xrightarrow{D}$ if $type_{\mathfrak{A}}(\rho) := (q, D, q')$ for some q and q'.

Define an equivalence relation $\sim_{\mathfrak{A}}$ on $\mathfrak{A}$-runs:

$$\rho_1 \sim_{\mathfrak{A}} \rho_2 \text{ if and only if } type_{\mathfrak{A}}(\rho_1) = type_{\mathfrak{A}}(\rho_2)$$

Weight Let D be a subset of the base B of $\mathfrak{A}$. The weight of D is defined as the cardinality of $B \setminus D$. The weight of a transition of $\mathfrak{A}$ is defined as follows. The weight of a successor transition is 0; the weight of limit transitions $(D, q) \in \delta_{\text{lim}}$ and $(q, D) \in \delta_{\text{lim}}$ is the weight of D. The weight of a run ρ is defined as the maximum of the weights of transitions that appears in ρ. We denote the weight of ρ by $\text{weight}(\rho)$; the weight is always between 0 and the cardinality of the base of $\mathfrak{A}$.

Lemma 9.1 (Main) *Assume that ρ is a run of a finite base automaton $\mathfrak{A}$.*

1. *If $\rho :\xrightarrow{D}$ and $\text{weight}(\rho) = \text{weight}(D) = w$, then there is a run on a chain of rank $\leq 2w + 1$, which is equivalent to ρ.*

2. Any run of weight $\leq w$ is equivalent to a run on a chain of rank $\leq 2w+2$.

As a consequence, we obtain the following small rank property:

Proposition 9.2 (small rank property) *Let $\mathfrak{A}$ be a finite base automaton with base of size $n_{\mathfrak{A}}$. Every run of $\mathfrak{A}$ is equivalent to a run on a chain of rank $\leq 2n_{\mathfrak{A}}+2$. In particular, if $\mathfrak{A}$ has an accepting run, then it accepts a chain of rank $\leq 2n_{\mathfrak{A}}+2$.*

The complexity analysis of our PSPACE algorithm was based on Lemma 7.2. Now we are ready to prove it.

Proof. (of Lemma 7.2.) Let Φ be a finite set of first-order formulas. By Theorem 8.3, every formula in $\varphi \in \Phi$ is equivalent to a finite-base automaton $\mathfrak{A}_\varphi$. Let n_Φ be an upper bound on the base size of $\mathfrak{A}_\varphi$ for $\varphi \in \Phi$.

Let $\psi = \varphi_1(\overline{P_1}) \wedge \cdots \wedge \varphi_k(\overline{P_k})$ be a Φ-conjunctive formula. By Lemma 8.2, ψ is equivalent to a finite base automata with the base of size $\leq k \times n_\Phi$. By Proposition 9.2, if ψ is satisfiable then it is satisfiable on a chain of rank $\leq k(2n_\Phi+2)$. Hence, we can define r_Φ as $(2n_\Phi+2)$. □

It is instructive to compare small rank property of finite base automata with short run property of simple ordinal automata from [7]. A simple ordinal automaton is a finite base automaton with $\delta_{\lim} \subseteq \mathcal{P}(B) \times Q$. Hence, the domain of every run ρ of a simple ordinal automaton is order-isomorphic to an ordinal, and if ρ is a run on $\mathcal{M}$ then $\mathcal{M}$ is a chain over an ordinal. An ordinal α has rank $i \geq 1$ iff $\alpha < \omega^{i+1}$. Lemma 6 in [7] states that every run of a simple ordinal automaton $\mathfrak{A}$ is equivalent to an $\mathfrak{A}$-run on an ordinal $< \omega^{n_{\mathfrak{A}}+1}$, where $n_{\mathfrak{A}}$ is the size of the base of $\mathfrak{A}$.

10 Conclusion, Further and Related Results

We provided an EXPTIME algorithm for the satisfiability problem for any temporal or modal logic with a finite set of ∃-MSO definable modalities over a recursively defined class of structures, and proved that EXPTIME-bound is optimal in the worst case.

Let *TL* be any temporal logic with a finite set of ∃-MSO definable modalities. We proved that the satisfiability problem for *TL* over the class of all linear orders can be solved in PSPACE. This improves the Cristau result [6] that the satisfiability problem over this class for the temporal logic having the four modalities Until, Since, Until_{Stavi} and Since_{Stavi} is in double exponential space, and implies Reynolds's conjecture.

In the rest of this section we explain how the PSPACE bound can be extended uniformly to many interesting classes of linear orders.

Let ψ be an ∃-MSO sentence. A set $\mathcal{C}$ of chains is said to be *definable by ψ*, if $\mathcal{C} = \{\mathcal{M} \mid \mathcal{M} \models \psi\}$. A set $\mathcal{C}$ of chains is said to be *definable by ψ relatively to a class* $\mathcal{C}'$, if $\mathcal{C} = \{\mathcal{M} \in \mathcal{C}' \mid \mathcal{M} \models \psi\}$.

Theorem 7.3 immediately implies

Corollary 10.1 *Let TL be a temporal logic with a finite set of ∃-MSO definable modalities, and let ψ be an ∃-MSO sentence. If the satisfiability problem for TL over $\mathcal{C}'$ is in* PSPACE*, then the satisfiability problem for TL over the class of chains definable by ψ relatively to $\mathcal{C}'$ is in* PSPACE*. In particular, the satisfiability problem for TL over the class of chains definable by ψ is in* PSPACE*.*

A linear order is called *unbounded* if it has neither a minimum nor a maximum; Note that an ∃-MSO formula φ is satisfiable in $\mathbb{Q}$ iff it is satisfiable in an unbounded dense order. There are first-order sentences *Unbound* and *Dense* that express that an order is unbounded, respectively, dense. Therefore, φ is satisfiable in $\mathbb{Q}$ iff $\mathit{Unbound} \wedge \mathit{Dense} \wedge \varphi$ is satisfiable over a linear order. Hence, there is a PSPACE algorithm for satisfiability in $\mathbb{Q}$.

Recall that a cut (L, U) of a linear order $\mathcal{L}$ is a gap if neither L has a maximal element, nor U has a minimal element and $L \neq \emptyset \neq U$. A chain is Dedekind-complete if its underlining order does not have gaps. The class of non-Dedekind chain can be easily definable by an ∃-MSO sentence. Hence, there is a PSPACE algorithm for the satisfiability over the class of non-Dedekind complete chains. The class of Dedekind complete chains is not definable by an ∃-MSO sentence. However, we will show (Theorem 10.7) that there is a PSPACE algorithm for the satisfiability over the class of Dedekind complete chains.

Let OP be a subset of $\{\omega, \omega^{-1}, \mathit{shuffle}\}$. Our proof can be easily modified to show the following variant of small rank property (Lemma 7.2).

Lemma 10.2 *For every finite set Φ of first-order formulas and every $OP \subseteq \{\omega, \omega^{-1}, \mathit{shuffle}\}$ there is $N_{\Phi,OP} \in \mathbb{N}$ such that every Φ-conjunctive formula ψ is satisfiable in $Cl(One, OP \cup \{+\})$ iff it is satisfiable in a chain $\mathcal{M} \in Cl(One, OP \cup \{+\})$ of rank $\leq |\psi| \times N_{\Phi,OP}$.*

Hence, the satisfiability problem for any temporal logic with a finite set of ∃-MSO definable modalities over $Cl(One, OP \cup \{+\})$ is in PSPACE.

Recall that a linear order is scattered if it does not contain a dense suborder (i.e., a substructure order-isomorphic to $\mathbb{Q}$). An ∃-MSO formula is satisfiable in a chain over an ordinal (respectively, over a scattered order) iff it is satisfiable in $Cl(One, \{\omega, +\}$ (respectively, in $Cl(One, \{\omega, \omega^{-1}, +\}$ [13, 19]. Hence, we obtain:

Theorem 10.3 *Let TL be a temporal logic with a finite set of modalities definable in the existential fragment of MSO.*

1. *The satisfiability problem for TL in the class of chains over ordinals is in* PSPACE *[7].*
2. *The satisfiability problem for TL in the class of scattered chains is in* PSPACE*.*

A linear order is *continuous* if it is dense and Dedekind-complete; it is separable if it has a countable dense subset. Any unbounded separable continuous order is order-isomorphic to the reals.

Burgess and Gurevich [5] proved that $TL(\mathsf{Until}, \mathsf{Since})$ is decidable over the reals. They introduced the following class of chains.

Definition 10.4 *Let $\mathcal{C}$ be the minimal class of chains that contains all one-element chains and has the following properties:*

1. *If $\mathcal{M}$ and $\mathcal{N}$ are in $\mathcal{C}$ and $\mathcal{M}$ has a maximum or $\mathcal{N}$ has a minimum, then $\mathcal{M} + \mathcal{N} \in \mathcal{C}$.*
2. *If $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M}$ has either a minimum or a maximum, then $\mathcal{M} \times \omega^{-1}$ and $\mathcal{M} \times \omega$ are in $\mathcal{C}$.*
3. *If $\mathcal{A} \subseteq \mathcal{C}$ is finite and each $\mathcal{M} \in \mathcal{A}$ has both a minimum and a maximum, and some $\mathcal{N} \in \mathcal{A}$ are one-element chains, then shuffle$(\mathcal{A}) \in \mathcal{C}$.*

The next theorem was a key step in their decidability proof.

Theorem 10.5 *Let φ be an $\exists$-MSO formula. The following are equivalent:*

1. *φ is satisfiable over the class of Dedekind-complete separable chains.*
2. *φ is satisfiable over the class of Dedekind-complete chains.*
3. *φ is satisfiable in $\mathcal{C}$.*

As a consequence, they obtained a (non-elementary) algorithm for the decidability of $TL(\mathsf{Until}, \mathsf{Since})$ over the reals.

The definition of $\mathcal{C}$ is slightly more general than the definition of a recursively defined class of structures. However, our definition is easily extended to the (mutual) recursive definition of a finite number of classes.

One can easily rephrase Definition 10.4 as a mutual recursive definition of three classes: $\mathcal{C}$, $\mathcal{C}_{\max}$ and $\mathcal{C}_{\min}$, where $\mathcal{C}_{\max}$ (respectively, $\mathcal{C}_{\min}$) is the set of chains in $\mathcal{C}$ with a maximal, (respectively, minimal) element. (Note that $\mathcal{C}_{\max}$ and $\mathcal{C}_{\min}$ are $\exists$-MSO definable relatively to $\mathcal{C}$.)

Our results are easily extended to these classes. In particular, for every finite set Φ of first-order formulas there is r_Φ such that a Φ-conjunctive formula ψ is satisfiable in $\mathcal{C}$ iff it is satisfiable in $\mathcal{M} \in \mathcal{C}$ of rank $\leq r_\Phi \times |\psi|$. Hence,

Lemma 10.6 *Let TL be a temporal logic with a finite set of modalities definable in $\exists$-MSO. The satisfiability problem for TL in $\mathcal{C}$ is in* PSPACE.

As a consequence, we obtain:

Theorem 10.7 *Let TL be a temporal logic with a finite set of modalities definable in the existential fragment of MSO.*

1. *The satisfiability problem for TL over the class of Dedekind-complete separable chains is in* PSPACE.
2. *The satisfiability problem for TL over the class of Dedekind-complete chains is in* PSPACE.
3. *The satisfiability problem for TL in the class of chains over the reals is in* PSPACE.
4. *The satisfiability problem for TL over the class of continuous chains is in* PSPACE.

Proof. (1) and (2) follow from Theorem 10.5 and Lemma 10.6.

Let *Unbound* and *Dense* be first-order formulas that express that an order is unbounded and dense. By Theorem 10.5, $\varphi \in TL$ is satisfiable over the reals iff $\varphi \wedge Dense \wedge Unbound$ is satisfiable in $\mathcal{C}$. Therefore, (3) follows by Lemma 10.6.

$\varphi \in TL$ is satisfiable over the class of continuous chains iff $\varphi \wedge Dense$ is satisfiable in $\mathcal{C}$. Therefore, (4) follows by Lemma 10.6. □

Similar arguments show that the satisfiability problem for *TL* over the classes of scattered Dedekind-complete chains, scattered non Dedekind-complete chains, and over many other classes is in PSPACE.

Reynolds [17] proved Theorem 10.7(3) for the temporal logic with two modalities Until and Since. Due to the Kamp theorem, this implies that the satisfiability problem over the reals for any temporal logic with a finite set of first-order definable modalities is in PSPACE. His proof relies on particular properties of Until and Since and uses temporal mosaics. The proofs in [17] are very non-trivial and difficult to grasp, probably because they have been developed from scratch.

We do not fully understand the Reynolds proof; however, there are some elements which are similar to our proof of Theorem 10.7(3). He considers operations on mosaics which correspond to sum, multiplication by ω and by ω^{-1} and shuffle of chains. He decides whether a finite set of small pieces is sufficient to be used to build a real-number model of a given formula. This is also equivalent to the existence of a winning strategy for player one in a two-player game played with mosaics. The search for a winning strategy is arranged into a search through a tree of mosaics. By establishing limits on the depth of the tree (a polynomial in terms of the length of the formula) he constructs a PSPACE algorithm. There is an analogy between such mosaic trees and construction trees for chains of finite rank.

Finally, let us note that the results of this paper hold for temporal logics with modalities having generalized truth tables definable by automata. Let $\mathfrak{A}$ be an automaton over the alphabet $\{0,1\}^{n+1}$. A modality O is said to be definable by $\mathfrak{A}$ if for every linear order $\mathcal{L} := \langle A, < \rangle$ and every $P_1, \dots, P_n \subseteq A$ there is a unique P such that $\langle A, <, P, P_1, \dots, P_n \rangle$ is accepted by $\mathfrak{A}$, moreover $P = O(P_1, \dots, P_n)$.

Theorem 10.8 *Let TL be a temporal logic with a finite set of modalities such that every modality is definable by an automaton. Then, the satisfiability problem for TL over the class of all chains is in* PSPACE.

References

1. N. Bedon, A. Bes, O. Carton, C. Rispal: Logic and Rational Languages of Words Indexed by Linear Orderings. CSR 2008: 76-85.
2. V. Bruyère and O. Carton. Automata on linear orderings.MFCS01: 236-247.
3. J. R. Büchi. On a decision method in restricted second order arithmetic. In: Logic, Methodology and Philosophy of Science, Stanford University Press, (1962), 1-11.
4. J. R. Büchi and D. Siefkes, The Monadic Second-order Theory of all Countable Ordinals, Springer Lecture Notes 328, 1973.

5. J. P. Burgess and Y. Gurevich. The decision problem for linear temporal logic. Notre Dame J. Formal Logic, 26(2):115-128, 1985.
6. J. Cristau. Automata and temporal logic over arbitrary linear time. FSTTCS 2009, 133-144.
7. S. Demri and A. Rabinovich. The Complexity of Temporal Logic with Until and Since over Ordinals. LPAR 2007: 531-545.
8. S. Feferman and R. L. Vaught. The first-order properties of products of algebraic systems. *Fundamenta Mathematicae* 47:57–103, 1959.
9. D. M. Gabbay, I. Hodkinson, M. Reynolds. Temporal Logics volume 1. Clarendon Press, Oxford 1994.
10. D. M. Gabbay, A. Pnueli, S. Shelah, J. Stavi. On the Temporal Analysis of Fairness. 7th POPL, pp. 163-173, 1980.
11. Y. Gurevich. Monadic second-order theories. In *Model-Theoretic Logics*, (J. Barwise and S. Feferman, eds.), 479-506, Springer-Verlag, 1985.
12. H. Kamp. Tense Logic and the Theory of Linear Order. Ph.D. thesis, University of California L.A., 1968.
13. H. Läuchli and J. Leonard. On the elementary theory of linear order. Fundamenta Mathematicae, 59:109-116, 1966.
14. M. O. Rabin. Decidability of second-order theories and automata on infinite trees. Transactions of the American Mathematical Society, vol. 141:1-35, 1969.
15. A. Rabinovich. Temporal logics over linear time domains are in PSPACE. Manuscript, 2009.
16. M. Reynolds. The complexity of the temporal logic with until over general linear time. J. Comput. Syst. Sci., 66:393-426, 2003.
17. M. Reynolds. The Complexity of Temporal Logic over the Reals. In the Annals of Pure and Applied Logic, 161(8): 1063-1096, 2010.
18. C. Rispal, O. Carton. Complementation of rational sets on countable scattered linear orderings. Int. J. Found. Comput. Sci. 16(4): 767-786, 2005.
19. J. G. Rosenstein. *Linear ordering.* Academic Press, New York, 1982.
20. S. Shelah. The monadic theory of order. *Ann. of Math.*, **102**, pp 349-419, 1975.
21. A. P. Sistla and E. M. Clarke. The Complexity of Propositional Linear Temporal Logics J. ACM 32(3): 733-749, 1985.
22. W. Thomas. Ehrenfeucht games, the composition method, and the monadic theory of ordinal words. In *Structures in Logic and Computer Science: A Selection of Essays in Honor of A. Ehrenfeucht*, LNCS 1261, pp. 118-143, 1997.