Canonical Models and the Complexity of Modal Team Logic

Martin Lück
Institut für Theoretische Informatik, Leibniz Universität Hannover
Appelstraße 4, 30167 Hannover, Germany
lueck@thi.uni-hannover.de

Abstract

We study modal team logic MTL, the team-semantical extension of classical modal logic closed under Boolean negation. Its fragments, such as modal dependence, independence, and inclusion logic, are well-understood. However, due to the unrestricted Boolean negation, the satisfiability problem of full MTL has been notoriously resistant to a complexity theoretical classification.

In our approach, we adapt the notion of canonical models for team semantics. By construction of such a model, we reduce the satisfiability problem of MTL to simple model checking. Afterwards, we show that this method is optimal in the sense that MTL-formulas can efficiently enforce canonicity.

Furthermore, to capture these results in terms of computational complexity, we introduce a non-elementary complexity class, TOWER(poly), and prove that the satisfiability and validity problem of MTL are complete for it. We also show that the fragments of MTL with bounded modal depth are complete for the levels of the elementary hierarchy (with polynomially many alternations).

2012 ACM Subject Classification Theory of computation → Complexity theory and logic, Theory of computation → Logic

Keywords and phrases team semantics, modal logic, complexity, satisfiability

Digital Object Identifier 10.4230/LIPIcs.CSL.2018.30

Related Version A full version of the paper is available at [30], `https://arxiv.org/abs/1709.05253`.

Acknowledgements The author wishes to thank Heribert Vollmer, Irena Schindler and Arne Meier, as well as the anonymous referees, for numerous helpful comments and hints.

1 Introduction

It is well-known that non-linear quantifier dependencies, such as w depending only on z in the sentence $\forall x\,\exists y\,\forall z\,\exists w\,\varphi$, cannot be expressed in first-order logic. To overcome this restriction, logics of incomplete information such as *independence-friendly logic* [19] have been studied. Later, Hodges [20] introduced *team semantics* to provide these logics with a compositional interpretation. The fundamental idea is to not consider only plain assignments to free variables, but instead whole sets of assignments, called *teams*.

In this vein, Väänänen [38] expressed non-linear quantifier dependencies by the *dependence atom* $=\!(x_1, \ldots, x_n, y)$, which intuitively states that the values of y in the team must depend only on those of $x_1, \ldots, x_n$. Logics with numerous other non-classical atoms such as *independence* $\perp$ [9], *inclusion* $\subseteq$ and *exclusion* $\mid$ [7] have been studied since, and have found manifold application in scientific areas such as statistics, database theory, physics, cryptography and social choice theory (see also Abramsky et al. [1]).

27th EACSL Annual Conference on Computer Science Logic (CSL 2018).
Editors: Dan Ghica and Achim Jung; Article No. 30; pp. 30:1–30:23
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 Complexity landscape of propositional and modal logics of dependence (∗DL), independence (∗IL), inclusion (∗Inc) and team logic (∗TL). Entries are completeness results unless stated otherwise.

Logic	Satisfiability	Validity	References
PDL	NP	NEXPTIME	[26, 36]
MDL	NEXPTIME	NEXPTIME	[33, 11]
PIL	NP	NEXPTIME-hard, in Π_2^E	[13]
MIL	NEXPTIME	Π_2^E-hard	[23, 10]
PInc	EXPTIME	co-NP	[13]
MInc	EXPTIME	co-NEXPTIME-hard	[16]
PTL	ATIME-ALT(exp, poly)	ATIME-ALT(exp, poly)	[12, 14]
MTL_k	ATIME-ALT($\exp_{k+1}$, poly)	ATIME-ALT($\exp_{k+1}$, poly)	Theorem 6.1
MTL	TOWER(poly)	TOWER(poly)	Theorem 6.1

Team semantics have also been adapted to a range of propositional [39, 12], modal [35], and temporal logics [25]. Not only have *propositional dependence logic* PDL [39] and *modal dependence logic* MDL [35] been extensively studied, but propositional and modal logics of independence and inclusion as well [23, 13, 18, 11]. Here, the non-classical atoms, such as the dependence atom, range over whole formulas. For example, the instance $=(p_1, \ldots, p_n, \Diamond\mathsf{unsafe})$ of a modal dependence atom may specify that the reachability of an unsafe state depends on an "access code" $p_1 \cdots p_n$ (and on nothing else), but instead of exhibiting the explicit function in question, it only stipulates the existence of such.

Most team logics lack a Boolean negation, and adding it as a connective $\sim$ usually increases both the expressive power and the complexity tremendously. The respective extensions of propositional and modal logic are called *propositional team logic* PTL [12, 40, 14] and *modal team logic* MTL [31, 22]. By means of the negation $\sim$, these logics can express all the non-classical atoms mentioned above, and in fact are expressively complete for their respective class of models [22, 40]. For these reasons, they are both interesting and natural logics.

The expressive power of MTL is well-understood [22], and a complete axiomatization was presented by the author [27]. Yet the complexity of the satisfiability problem has been an open question [31, 22, 6, 15]. Recently, certain fragments of MTL with restricted negation were shown ATIME-ALT(exp, poly)-complete using the well-known filtration method [28]. In the same paper, however, it was shown that no elementary upper bound for full MTL can be established by the same approach, whereas the best known lower bound is ATIME-ALT(exp, poly)-hardness, inherited from the fragment PTL [14]. Analogously, the best known model size lower bound is – as for ordinary modal logic – exponential in the size of the formula.

Contribution. We show that MTL is complete for a non-elementary class we call TOWER(poly), which contains, roughly speaking, the problems decidable in a runtime that is a tower of nested exponentials with polynomial height. Likewise, we show that the fragments MTL_k of bounded modal depth k are complete for a class we call ATIME-ALT($\exp_{k+1}$, poly) and which corresponds to $(k + 1)$-fold exponential runtime and polynomially many alternations. These results fill a long-standing gap in the active field of propositional and modal team logics (see Table 1).

In our approach, we consider *canonical* or *universal* models. Loosely speaking, a canonical model satisfies every satisfiable formula in some of its submodels, and such models have been long known for, e.g., many systems of modal logic [2]. In Section 3, we adapt this notion for modal logics with team semantics, and prove that such models exist for MTL. This enables us to reduce the satisfiability problem to simple model checking, albeit on models that are of non-elementary size with respect to $|\Phi| + k$, where Φ are the available propositional variables and k is a bound on the modal depth.

Nonetheless, this approach is essentially optimal: In Section 4 and 5, we show that MTL can, in a certain sense, *efficiently enforce* canonical models, that is, with formulas that are of size polynomial in $|\Phi|+k$. In this vein, we then obtain the matching complexity lower bounds in Section 6 by encoding computations of non-elementary length in such large models.

To the author's best knowledge, the classes $\text{ATIME-ALT}(\exp_k, \text{poly})$ and $\text{TOWER}(\text{poly})$ have not explicitly been considered before. However, there are several candidates for other natural complete problems. More precisely, there exist problems in TOWER(poly) that are provably non-elementary, such as the satisfiability problem of separated first-order logic [37], the equivalence problem for star-free expressions [34], or the first-order theory of finite trees [4], to only name a few.

Another example is the two-variable fragment of first-order team logic, $\mathsf{FO}^2(\sim)$. It is related to MTL in the same fashion as classical two-variable logic FO^2 to ML. Due to a reduction from MTL to $\mathsf{FO}^2(\sim)$ (see [29]), the satisfiability and validity problems of $\mathsf{FO}^2(\sim)$ are TOWER(poly)-complete problems as a corollary of this paper, while its fragments $\mathsf{FO}^2_k(\sim)$ of bounded quantifier rank k are $\text{ATIME-ALT}(\exp_{k+1}, \text{poly})$-hard.

Due to space constraints, several technical proofs (which are marked with $(\star)$) are omitted or only sketched. They can be found in the full version of this paper [30].

2 Preliminaries

The power set of a set X is $\mathfrak{P}(X)$. We let $|X|$ denote the length of the encoding of a formula or structure X. The sets of all satisfiable resp. valid formulas of a given logic $\mathcal{L}$ are $\mathsf{SAT}(\mathcal{L})$ and $\mathsf{VAL}(\mathcal{L})$, respectively.

We assume the reader to be familiar with alternating Turing machines [3]. We assume all reductions in this paper implicitly as logspace reductions $\leq_m^{\log}$.

The class $\text{ATIME-ALT}(\exp, \text{poly})$ contains the problems decidable by an alternating Turing machine in time $2^{p(n)}$ with $p(n)$ alternations, for a polynomial p. It is a natural class that has several complete problems [13, 21, 14]. Here, we generalize it to capture the *elementary hierarchy* $\exp_k(n)$, defined by $\exp_0(n) := n$ and $\exp_{k+1}(n) := 2^{\exp_k(n)}$.

▶ **Definition 2.1.** For $k \geq 0$, $\text{ATIME-ALT}(\exp_k, \text{poly})$ is the class of problems decided by an alternating Turing machine with at most $p(n)$ alternations and runtime at most $\exp_k(p(n))$, for a polynomial p.

Note that setting $k = 0$ or $k = 1$ yields the classes PSPACE and $\text{ATIME-ALT}(\exp, \text{poly})$, respectively [3]. If k is replaced by a polynomial instead, we obtain the following class.

▶ **Definition 2.2.** TOWER(poly) is the class of problems that are decided by a deterministic Turing machine in time $\exp_{p(n)}(1)$ for some polynomial p.

Note that a similar class, TOWER, is defined by replacing p by an arbitrary elementary function [32]. By contrast, to the author's best knowledge, TOWER(poly) has not yet been explicitly studied. The reader may verify that both $\text{ATIME-ALT}(\exp_k, \text{poly})$ and TOWER(poly) are closed under polynomial time reductions (and hence also $\leq_m^{\log}$).

Modal team logic

We fix a countably infinite set $\mathcal{PS}$ of propositional symbols. *Modal team logic* MTL, introduced by Müller [31], extends classical modal logic ML as in the following grammar, where φ denotes an MTL-formula, α an ML-formula, and $p \in \mathcal{PS}$.

$$\varphi ::= {\sim}\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \alpha$$
$$\alpha ::= \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Box\alpha \mid \Diamond\alpha \mid p \mid \top$$

The set of propositional variables occurring in $\varphi \in \mathsf{MTL}$ is denoted by $\mathsf{Prop}(\varphi)$.

We use the common abbreviations $\bot := \neg\top$, $\alpha \to \beta := \neg\alpha \vee \beta$ and $\alpha \leftrightarrow \beta := (\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta)$. For easier distinction, we have classical formulas denoted by $\alpha, \beta, \gamma, \ldots$ and reserve $\varphi, \psi, \vartheta, \ldots$ for general team-logical formulas.

The modal depth $\mathsf{md}(\theta)$ of an (ML or MTL) formula θ is recursively defined:

$$\begin{aligned}
\mathsf{md}(p) &:= \mathsf{md}(\top) &&:= 0\\
\mathsf{md}({\sim}\varphi) &:= \mathsf{md}(\neg\varphi) &&:= \mathsf{md}(\varphi)\\
\mathsf{md}(\varphi \wedge \psi) &:= \mathsf{md}(\varphi \vee \psi) &&:= \max\{\mathsf{md}(\varphi), \mathsf{md}(\psi)\}\\
\mathsf{md}(\Diamond\varphi) &:= \mathsf{md}(\Box\varphi) &&:= \mathsf{md}(\varphi) + 1
\end{aligned}$$

ML_k and MTL_k are the fragments of ML and MTL with modal depth $\leq k$, respectively. If the propositions are restricted to a fixed set $\Phi \subseteq \mathcal{PS}$ as well, then the fragment is denoted by ML_k^Φ, or MTL_k^Φ, respectively.

Let $\Phi \subseteq \mathcal{PS}$ be a finite set of propositions. A *Kripke structure* (over Φ) is a tuple $\mathcal{K} = (W, R, V)$, where W is a set of *worlds*, (W, R) is a directed graph, and $V : \Phi \to \mathfrak{P}(W)$ is the *valuation*. Occasionally, by slight abuse of notation, we use the mapping $V^{-1} : W \to \mathfrak{P}(\Phi)$ defined by $V^{-1}(w) := \{p \in \Phi \mid w \in V(p)\}$ instead of V, i.e., the set of propositions that are true in a given world.

If $w \in W$, then $(\mathcal{K}, w)$ is called *pointed structure*. ML is evaluated on pointed structures in the classical Kripke semantics. By contrast, MTL is evaluated on pairs $(\mathcal{K}, T)$, called *structures with teams*, where $T \subseteq W$ is called *team* (in $\mathcal{K}$).

Every team T has an *image* $RT := \{v \mid w \in T, (w, v) \in R\}$, and if $w \in W$, we simply write Rw instead of $R\{w\}$. R^iT is inductively defined as $R^0T := T$ and $R^{i+1}T := RR^iT$. A *successor team* of T is a team S such that $S \subseteq RT$ and $T \subseteq R^{-1}S$, where $R^{-1} := \{(v, w) \mid (w, v) \in R\}$. Intuitively, S is formed by picking at least one successor of every world in T.

The semantics of MTL can now be defined as follows.[1]

$$\begin{aligned}
(\mathcal{K}, T) \vDash \alpha &\Leftrightarrow \forall w \in T : (\mathcal{K}, w) \vDash \alpha \text{ if } \alpha \in \mathsf{ML}, \text{ and otherwise as}\\
(\mathcal{K}, T) \vDash {\sim}\psi &\Leftrightarrow (\mathcal{K}, T) \nvDash \psi,\\
(\mathcal{K}, T) \vDash \psi \wedge \theta &\Leftrightarrow (\mathcal{K}, T) \vDash \psi \text{ and } (\mathcal{K}, T) \vDash \theta,\\
(\mathcal{K}, T) \vDash \psi \vee \theta &\Leftrightarrow \exists S, U \subseteq T \text{ such that } T = S \cup U,\ (\mathcal{K}, S) \vDash \psi, \text{ and } (\mathcal{K}, U) \vDash \theta,\\
(\mathcal{K}, T) \vDash \Diamond\psi &\Leftrightarrow (\mathcal{K}, S) \vDash \psi \text{ for some successor team } S \text{ of } T,\\
(\mathcal{K}, T) \vDash \Box\psi &\Leftrightarrow (\mathcal{K}, RT) \vDash \psi.
\end{aligned}$$

We often omit $\mathcal{K}$ and write $T \vDash \varphi$ or $w \vDash \alpha$.

[1] Often, the "atoms" of MTL are restricted to literals $p, \neg p$ instead of ML-formulas α. However, this implies a restriction to formulas in negation normal form, and both definitions are equivalent due to the *flatness* property of ML (cf. [22, Proposition 2.2]).

An MTL-formula φ is *satisfiable* if it is true in some structure with team over $\mathsf{Prop}(\varphi)$, which is then called a *model* of φ. Analogously, φ is *valid* if it is true in every structure with team over $\mathsf{Prop}(\varphi)$.

Note that the empty team is usually excluded in the above definition, since most $\sim$-free logics with team semantics have the *empty team property*, i.e., the empty team trivially satisfying every formula [35, 23, 18]. However, this distinction is unnecessary for MTL: φ is satisfiable iff $\top \vee \varphi$ is true in some non-empty team[2], and φ is true in some non-empty team iff $\sim\bot \wedge \varphi$ is satisfiable.

The modality-free fragment MTL_0 syntactically coincides with *propositional team logic* PTL [12, 14, 40]. The usual interpretations of the latter, i.e., sets of Boolean assignments, can easily be represented as teams in Kripke structures. For this reason, we identify PTL and MTL_0 in this paper.

Note that the connectives $\vee$, $\rightarrow$ and $\neg$ are not the usual truth-functional connectives on the level of teams, i.e., Boolean disjunction, implication and negation. The exception are singleton teams, on which team semantics and Kripke semantics coincide. Using $\wedge$ and $\sim$ however, we can define Boolean disjunction $\varphi_1 \owedge \varphi_2 := \sim(\sim\varphi_1 \wedge \sim\varphi_2)$ and implication $\varphi_1 \multimap \varphi_2 := \sim\varphi_1 \owedge \varphi_2$.

The notation $\Box^i\varphi$ is defined via $\Box^0\varphi := \varphi$ and $\Box^{i+1}\varphi := \Box\Box^i\varphi$, and analogously for $\Diamond^i\varphi$. To state that at least one element of a team satisfies $\alpha \in \mathsf{ML}$, we write $\mathsf{E}\alpha := \sim\neg\alpha$. That the truth value of α is constant in the team is expressed by the *constancy atom* $=\!(\alpha) := \alpha \owedge \neg\alpha$.

The well-known *bisimulation* relation $\rightleftharpoons_k^\Phi$ fundamentally defines the expressive power of modal logic [2] and plays a key role in our results.

▶ **Definition 2.3.** Let $\Phi \subseteq \mathcal{PS}$ and $k \geq 0$. For $i \in \{1,2\}$, let $(\mathcal{K}_i, w_i)$ be a pointed structure, where $\mathcal{K}_i = (W_i, R_i, V_i)$. Then $(\mathcal{K}_1, w_1)$ and $(\mathcal{K}_2, w_2)$ are (Φ, k)-*bisimilar*, in symbols $(\mathcal{K}_1, w_1) \rightleftharpoons_k^\Phi (\mathcal{K}_2, w_2)$, if

- $\forall p \in \Phi\colon w_1 \in V_1(p) \Leftrightarrow w_2 \in V_2(p)$,
- and if $k > 0$,
  - $\forall v_1 \in R_1 w_1\colon \exists v_2 \in R_2 w_2\colon (\mathcal{K}_1, v_1) \rightleftharpoons_{k-1}^\Phi (\mathcal{K}_2, v_2)$ (*forward* condition),
  - $\forall v_2 \in R_2 w_2\colon \exists v_1 \in R_1 w_1\colon (\mathcal{K}_1, v_1) \rightleftharpoons_{k-1}^\Phi (\mathcal{K}_2, v_2)$ (*backward* condition).

The notion of bisimulation was also lifted to team semantics by Hella et al. [17]:

▶ **Definition 2.4** (cf. [17, 23, 22]). Let $\Phi \subseteq \mathcal{PS}$ and $k \geq 0$. For $i \in \{1,2\}$, let $(\mathcal{K}_i, T_i)$ be a structure with team. Then $(\mathcal{K}_1, T_1)$ and $(\mathcal{K}_2, T_2)$ are (Φ, k)-*team-bisimilar*, written $(\mathcal{K}_1, T_1) \rightleftharpoons_k^\Phi (\mathcal{K}_2, T_2)$, if

- $\forall w_1 \in T_1\colon \exists w_2 \in T_2\colon (\mathcal{K}_1, w_1) \rightleftharpoons_k^\Phi (\mathcal{K}_2, w_2)$,
- $\forall w_2 \in T_2\colon \exists w_1 \in T_1\colon (\mathcal{K}_1, w_1) \rightleftharpoons_k^\Phi (\mathcal{K}_2, w_2)$.

If no confusion can arise, we will also refer to teams T_1, T_2 that are (Φ, k)-team-bisimilar simply as (Φ, k)-bisimilar. The proofs of the following propositions are straightforward and can be found in the full version [30].

▶ **Proposition 2.5** ($\star$). *Let $\Phi \subseteq \mathcal{PS}$ be finite, and $k \geq 0$. For $i \in \{1,2\}$, let $(\mathcal{K}_i, w_i)$ be a pointed structure, where $\mathcal{K}_i = (W_i, R_i, V_i)$. Then the following statements are equivalent:*

1. $\forall \alpha \in \mathsf{ML}_k^\Phi\colon (\mathcal{K}_1, w_1) \vDash \alpha \Leftrightarrow (\mathcal{K}_2, w_2) \vDash \alpha$,
2. $(\mathcal{K}_1, w_1) \rightleftharpoons_k^\Phi (\mathcal{K}_2, w_2)$,

[2] In team semantics, $\top \vee \varphi$ is not tautologically true, but rather existentially quantifies a subteam.

3. $(\mathcal{K}_1, \{w_1\}) \rightleftharpoons_k^\Phi (\mathcal{K}_2, \{w_2\})$.

Moreover, if $k > 0$*, they are equivalent to:*

4. $(\mathcal{K}_1, w_1) \rightleftharpoons_0^\Phi (\mathcal{K}_2, w_2)$ *and* $(\mathcal{K}_1, R_1 w_1) \rightleftharpoons_{k-1}^\Phi (\mathcal{K}_2, R_2 w_2)$.

As a result, the *forward* and *backward* condition from Definition 2.3 can be equivalently stated in terms of team-bisimilarity of the respective images. On the level of teams, a similar characterization holds:

▶ **Proposition 2.6** ($\star$). *Let* $\Phi \subseteq \mathcal{PS}$ *be finite, and* $k \geq 0$*. Let* $(\mathcal{K}_i, T_i)$ *be a structure with team for* $i \in \{1, 2\}$*. Then the following statements are equivalent:*

1. $\forall \alpha \in \mathsf{ML}_k^\Phi : (\mathcal{K}_1, T_1) \vDash \alpha \Leftrightarrow (\mathcal{K}_2, T_2) \vDash \alpha$,

2. $\forall \varphi \in \mathsf{MTL}_k^\Phi : (\mathcal{K}_1, T_1) \vDash \varphi \Leftrightarrow (\mathcal{K}_2, T_2) \vDash \varphi$,

3. $(\mathcal{K}_1, T_1) \rightleftharpoons_k^\Phi (\mathcal{K}_2, T_2)$,

3 Types and canonical models

Many modal logics admit a "universal" model, also called *canonical model*. Given a canonical model $\mathcal{K}$, and a satisfiable formula (or set of formulas), the latter is then also true in some point of $\mathcal{K}$. See also Blackburn et al. [2, Section 4.2] for the explicit construction of such a model for ML.

Unfortunately, a canonical model for ML is necessarily infinite, and consequently impractical for complexity theoretic considerations. Instead, we define (Φ, k)-*canonical models* for finite $\Phi \subseteq \mathcal{PS}$ and $k \in \mathbb{N}$, which are then proved canonical for the fragment ML_k^Φ. However, by Proposition 2.5, the size of a (Φ, k)-canonical model is necessarily at least the number of equivalence classes of $\rightleftharpoons_k^\Phi$.

The equivalence classes of $\rightleftharpoons_k^\Phi$ are proper classes. However, speaking about teams would require *sets* of such classes. For this reason, we inductively define *types*, which properly reflect bisimulation, but exist as sets. We usually refer to types as τ.

▶ **Definition 3.1.** Let $\Phi \subseteq \mathcal{PS}$ be finite. The set of (Φ, k)-*types*, written Δ_k^Φ, is defined inductively as $\Delta_0^\Phi := \mathfrak{P}(\Phi) \times \{\emptyset\}$ and $\Delta_{k+1}^\Phi := \mathfrak{P}(\Phi) \times \mathfrak{P}(\Delta_k^\Phi)$.

Let $(\mathcal{K}, w) = (W, R, V, w)$ be a pointed structure. Then its (Φ, k)-type, written $[\![\mathcal{K}, w]\!]_k^\Phi$, is the unique $(\Phi', \Delta') \in \Delta_k^\Phi$ such that $V^{-1}(w) = \Phi'$ and, in case $k > 0$, additionally $\forall \tau' \in \Delta_{k-1}^\Phi : \tau' \in \Delta' \Leftrightarrow \exists v \in Rw : [\![\mathcal{K}, v]\!]_{k-1}^\Phi = \tau'$.

Given a team T in $\mathcal{K}$, the types in T are denoted by $[\![\mathcal{K}, T]\!]_k^\Phi := \{[\![\mathcal{K}, w]\!]_k^\Phi \mid w \in T\}$.

For a type $\tau = (\Phi', \Delta')$, we define shorthands $\Phi_\tau := \Phi'$ and $\mathcal{R}\tau := \Delta'$.

Intuitively, the first component Φ_τ consists of the propositions which any model of type τ must satisfy in its root, and $\mathcal{R}\tau$ is the set of types which any model of type τ must contain in the image of its root. Roughly speaking, Φ_τ reflects the first condition of Definition 2.3, propositional equivalence, while $\mathcal{R}\tau$ reflects the forward and backward conditions.

Every type $\tau \in \Delta_k^\Phi$ is satisfiable in the sense that there is at least one pointed structure $(\mathcal{K}, w)$ such that $[\![\mathcal{K}, w]\!]_k^\Phi = \tau$.

The following assertions are straightforward to prove by induction, and ascertain that types properly reflect the notion of bisimulation.

▶ **Proposition 3.2** ($\star$). *Let* $\Phi \subseteq \mathcal{PS}$ *be finite and* $k \geq 0$*. Then* $(\mathcal{K}, w) \rightleftharpoons_k^\Phi (\mathcal{K}', w')$ *if and only if* $[\![\mathcal{K}, w]\!]_k^\Phi = [\![\mathcal{K}', w']\!]_k^\Phi$*, and* $(\mathcal{K}, T) \rightleftharpoons_k^\Phi (\mathcal{K}', T')$ *if and only if* $[\![\mathcal{K}, T]\!]_k^\Phi = [\![\mathcal{K}', T']\!]_k^\Phi$.

We are now ready to state the formal definition of canonicity:

▶ **Definition 3.3.** A structure with team $(\mathcal{K}, T)$ is (Φ, k)-*canonical* if $[\![\mathcal{K}, T]\!]_k^\Phi = \Delta_k^\Phi$.

In the following, we often omit Φ and $\mathcal{K}$ and write only $[\![w]\!]_k$ or $[\![T]\!]_k$, and simply say that T is (Φ, k)-canonical if $\mathcal{K}$ is clear.

It is a standard result that for every Φ and $k \geq 0$ there exists a (Φ, k)-canonical model (cf. Blackburn et al. [2]), or in other words, that the logic ML_k^Φ admits canonical models.

Canonical models in team semantics

The logic MTL is significantly more expressive than ML [22]. Nonetheless, we will show that every satisfiable MTL_k^Φ-formula can be satisfied in a (Φ, k)-canonical model. In other words, the canonical models of MTL_k^Φ and ML_k^Φ actually coincide.

▶ **Theorem 3.4.** *Let $(\mathcal{K}, T)$ be (Φ, k)-canonical and $\varphi \in \mathsf{MTL}_k^\Phi$. Then φ is satisfiable if and only if $(\mathcal{K}, T') \vDash \varphi$ for some $T' \subseteq T$.*

Proof. Assume $(\mathcal{K}, T)$ and φ are as above. As the direction from right to left is trivial, suppose that φ is satisfiable, i.e., has a model $(\hat{\mathcal{K}}, \hat{T})$. As a team in $\mathcal{K}$ that satisfies φ, we define

$$T' := \left\{ w \in T \;\middle|\; [\![\mathcal{K}, w]\!]_k^\Phi \in [\![\hat{\mathcal{K}}, \hat{T}]\!]_k^\Phi \right\}.$$

By Proposition 2.6 and 3.2, it suffices to prove $[\![\hat{\mathcal{K}}, \hat{T}]\!]_k^\Phi = [\![\mathcal{K}, T']\!]_k^\Phi$. Moreover, the direction "$\supseteq$" is clear by definition. As T is (Φ, k)-canonical, for every $\tau \in [\![\hat{\mathcal{K}}, \hat{T}]\!]_k^\Phi$ there exists a world $w \in T$ of type τ. Consequently, $[\![\hat{\mathcal{K}}, \hat{T}]\!]_k^\Phi \subseteq [\![\mathcal{K}, T']\!]_k^\Phi$. ◀

How large is a (Φ, k)-canonical model at least? The number of types can be written via the function $\exp_k^*$, which is defined by

$$\exp_0^*(n) := n, \qquad \exp_{k+1}^*(n) := n \cdot 2^{\exp_k^*(n)}.$$

Observe that this function resembles $\exp_k(n)$ (cf. p. 3) except for an additional factor of n in every "level" of the nested exponents. By Definition 3.1, we immediately obtain:

▶ **Proposition 3.5.** $|\Delta_k^\Phi| = \exp_k^*\left(2^{|\Phi|}\right)$ *for all $k \geq 0$ and finite $\Phi \subseteq \mathcal{PS}$.*

Next, we present an algorithm that solves the satisfiability and validity problems of MTL and its fragments MTL_k by computing a canonical model. Let us first explicate this construction in a lemma.

▶ **Lemma 3.6.** *There is an algorithm that, given $\Phi \subseteq \mathcal{PS}$ and $k \geq 0$, computes a (Φ, k)-canonical model in time polynomial in $|\Delta_k^\Phi|$.*

Proof. Let $\mathcal{K} = (W, R, V)$ be the computed structure. The idea is to construct sets $L_0 \cup L_1 \cup \cdots \cup L_k =: W$ of worlds in stage-wise manner such that L_i is (Φ, i)-canonical.

For L_0, we simply add a world w for each $\Phi' \in \mathfrak{P}(\Phi)$ such that $V^{-1}(w) = \Phi'$.

For $i > 0$, we iterate over all $L' \in \mathfrak{P}(L_{i-1})$ and $\Phi' \in \mathfrak{P}(\Phi)$ and insert a new world w into L_i such that $Rw = L'$ and again $V^{-1}(w) = \Phi'$. An inductive argument shows that L_i is (Φ, i)-canonical for all $i \in \{0, \ldots, k\}$. As $k \leq |\Delta_k^\Phi|$, and each L_i is constructed in time polynomial in $|\Delta_i^\Phi| \leq |\Delta_k^\Phi|$, the overall runtime is polynomial in $|\Delta_k^\Phi|$. ◀

The next lemma allows, roughly speaking, to replace a polynomial of $\exp_k^*$ by simply $\exp_k$, with only polynomial blowup in its argument.

▶ **Lemma 3.7.** *For every polynomial p there is a polynomial q such that $p(\exp_k^*(n)) \leq \exp_k(q((k+1)\cdot n))$ for all $k \geq 0$ and $n \geq 1$.*

Proof. For $p(n)$ bounded by cn^d, with $c, d \in \mathbb{N}$, let $q(n) := cdn^d + c$ (cf. [30]). ◀

▶ **Theorem 3.8.** $\mathsf{SAT}(\mathsf{MTL}_k)$ *and* $\mathsf{VAL}(\mathsf{MTL}_k)$ *are in* $\mathrm{ATIME\text{-}ALT}(\exp_{k+1}, \mathrm{poly})$.

Proof. Consider the following algorithm. Let $\varphi \in \mathsf{MTL}_k$ be the input, $n := |\varphi|$, and $\Phi := \mathsf{Prop}(\varphi)$. Construct deterministically, as in Lemma 3.6, a (Φ, k)-canonical structure $(\mathcal{K}, T) = (W, R, V, T)$ in time $p(|\Delta_k^\Phi|)$ for a polynomial p.

By a result of Müller [31], the model checking problem of MTL is solvable by an alternating Turing machine that has runtime polynomial in $|\varphi| + |\mathcal{K}|$, and alternations polynomial in $|\varphi|$. We call this algorithm as a subroutine: by Theorem 3.4, φ is satisfiable (resp. valid) if and only if for at least one team (resp. all teams) $T' \subseteq T$ we have $(\mathcal{K}, T') \vDash \varphi$. Equivalently, this is the case if and only if $(\mathcal{K}, T)$ satisfies $\top \vee \varphi$ (resp. $\sim(\top \vee \sim\varphi)$).

Let us turn to the overall runtime. $\mathcal{K}$ is constructed in time polynomial in $|\Delta_k^\Phi| = \exp_k^*(2^{|\Phi|}) \leq \exp_{k+1}^*(|\Phi|) \leq \exp_{k+1}^*(n)$. The subsequent model checking runs in time polynomial in $|\mathcal{K}| + n$, and hence polynomial in $\exp_{k+1}^*(n)$ as well. By Lemma 3.7, we obtain a total runtime of $\exp_{k+1}(q((k+2)\cdot n))$ for a polynomial q. ◀

The upper bound for MTL can be proved similarly, since $k := \mathsf{md}(\varphi)$ is polynomial in $|\varphi|$. Moreover, the alternations can be eliminated with additional exponential blowup.

▶ **Corollary 3.9.** $\mathsf{SAT}(\mathsf{MTL})$ *and* $\mathsf{VAL}(\mathsf{MTL})$ *are in* $\mathrm{TOWER}(\mathrm{poly})$.

4 Efficiently expressing bisimilarity

Kontinen et al. [22] proved that MTL is expressively complete up to bisimulation, i.e., it can define every property of teams that is closed under $\rightleftharpoons_k^\Phi$ for some finite Φ and k. Two such team properties are in fact (Φ, k)-bisimilarity itself – in the sense that two worlds in a team have the same type – as well as (Φ, k)-canonicity. Consequently, these properties are defined by MTL_k^Φ-formulas. However, by a simple counting argument, formulas defining arbitrary team properties are of non-elementary size w. r. t. Φ and k in the worst case.

From now on, we always assume some finite $\Phi \subseteq \mathcal{PS}$ and omit it in the notation, i.e., we write *k-canonicity*, *k-bisimilarity*, $\rightleftharpoons_k$, and so on.

In this section, we present an "approximation" (in a sense we clarify below) of k-bisimilarity that can be expressed in a formula χ_k that is of polynomial size in Φ and k. Likewise, in Section 5 we present a formula canon_k of polynomial size that expresses k-canonicity. Finally, in Section 6, we apply χ_k and canon_k in order to prove the lower bound for Corollary 3.9, i.e., $\mathrm{TOWER}(\mathrm{poly})$-hardness of $\mathsf{SAT}(\mathsf{MTL})$ and $\mathsf{VAL}(\mathsf{MTL})$ (and an analogous result for Theorem 3.8). Here, the idea is to enforce a sufficiently large structure with canon_k and then to encode a non-elementary computation into it. Clearly, χ_k and canon_k being polynomial in Φ and k is crucial for the reduction.

Scopes

To implement k-bisimilarity, we pursue a recursive approach. In the spirit of Proposition 2.5, the $(k+1)$-bisimilarity of two points w, v is expressed in terms of k-team-bisimilarity of Rw and Rv. Conversely, to verify k-team-bisimilarity of Rw and Rv, we proceed analogously to the *forward* and *backward* conditions of Definition 2.3 and reduce the problem to checking k-bisimilarity of pairs of points in Rw and Rv.

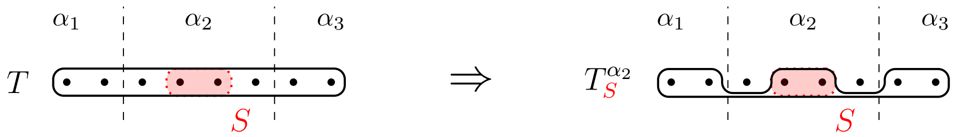

Figure 1 Example of subteam selection in the scope α_2.

A clear obstacle is that MTL cannot speak about two teams Rw, Rv simultaneously, let alone check for bisimilarity. Instead, we consider a team that is the "marked union" of Rw and Rv.

More generally, for all formulas $\alpha \in \mathsf{ML}$ we define the subteam $T_\alpha := \{\, w \in T \mid w \vDash \alpha \,\}$. The corresponding "decoding" operator

$$\alpha \hookrightarrow \varphi := \neg\alpha \vee (\alpha \wedge \varphi)$$

was considered by Kontinen and Nurmi [24] and Galliani [8]. Here, $\alpha \hookrightarrow \varphi$ is true in T if and only if $T_\alpha \vDash \varphi$.

Now, instead of defining an n-ary relation on teams, a formula φ can define a unary relation – a team property – parameterized by "marker formulas" $\alpha_1, \dots, \alpha_n \in \mathsf{ML}$. We emphasize this by writing $\varphi(\alpha_1, \dots, \alpha_n)$.

This is the "approximation" mentioned earlier: In order to compare Rw and Rv, we require that $Rw = T_\alpha$ and $Rv = T_\beta$ for some team T and distinct $\alpha, \beta \in \mathsf{ML}$. It will be useful if the "markers" are invariant under traversing edges in the structure:

▶ **Definition 4.1.** Let $\mathcal{K} = (W, R, V)$ be a Kripke structure. A formula $\alpha \in \mathsf{ML}$ is called a *scope (in $\mathcal{K}$)* if $(w, v) \in R$ implies $w \vDash \alpha \Leftrightarrow v \vDash \alpha$. Two scopes α, β are called *disjoint (in $\mathcal{K}$)* if W_α and W_β are disjoint.

In order to avoid interference, we always assume that scopes are formulas in $\mathsf{ML}_0^{\mathcal{PS}\setminus\Phi}$, i.e., they are always purely propositional and do not contain propositions from Φ.

It is desirable to be able to speak about subteams in a specific scope. Formally, if S is a team, let $T_S^\alpha := T_{\neg\alpha} \cup (T_\alpha \cap S)$. For singletons $\{w\}$, we simply write T_w^α instead of $T_{\{w\}}^\alpha$. Intuitively, T_S^α is obtained from T by "shrinking" the subteam T_α down to S without impairing $T \setminus T_\alpha$ (see Figure 1 for an example).

The following observations are straightforward:

▶ **Proposition 4.2** ([30]). *Let α, β be disjoint scopes and S, U, T teams in a Kripke structure $\mathcal{K} = (W, R, V)$. Then the following laws hold:*

1. *Distributive laws: $(T \cap S)_\alpha = T_\alpha \cap S = T \cap S_\alpha = T_\alpha \cap S_\alpha$ and $(T \cup S)_\alpha = T_\alpha \cup S_\alpha$.*
2. *Disjoint selection commutes: $\big(T_S^\alpha\big)_U^\beta = \big(T_U^\beta\big)_S^\alpha$.*
3. *Disjoint selection is independent: $\big((T_S^\alpha)_U^\beta\big)_\alpha = T_\alpha \cap S$.*
4. *Image and scope commute: $(RT)_\alpha = \big(R(T_\alpha)\big)_\alpha = R(T_\alpha)$.*
5. *Selection propagates: If $S \subseteq T$, then $R\big(T_S^\alpha\big) = (RT)_{RS}^\alpha$.*

Accordingly, we write $R^i T_\alpha$ instead of $(R^i T)_\alpha$ or $R^i(T_\alpha)$ and $T_{S_1,S_2}^{\alpha_1,\alpha_2}$ for $(T_{S_1}^{\alpha_1})_{S_2}^{\alpha_2}$.

Subteam quantifiers

We refer to the following abbreviations as *subteam quantifiers*, where $\alpha \in \mathsf{ML}$:

$$\exists_\alpha^{\subseteq}\, \varphi := \alpha \vee \varphi \qquad \forall_\alpha^{\subseteq}\, \varphi := {\sim}\exists_\alpha^{\subseteq}{\sim}\varphi$$

$$\exists_\alpha^{1}\, \varphi := \exists_\alpha^{\subseteq}\big[\mathsf{E}\alpha \wedge \forall_\alpha^{\subseteq}(\mathsf{E}\alpha \twoheadrightarrow \varphi)\big] \qquad \forall_\alpha^{1}\, \varphi := {\sim}\exists_\alpha^{1}{\sim}\varphi$$

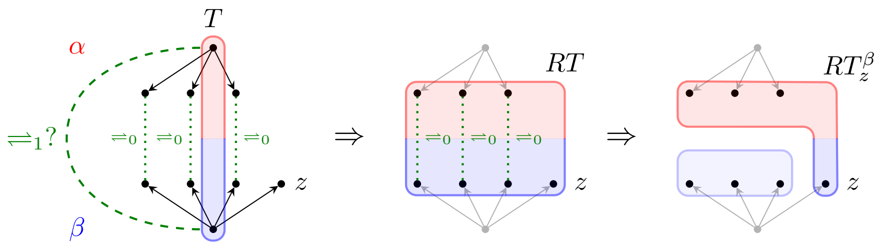

Figure 2 As z violates the *backward* condition, $\chi_0^*(\alpha,\beta)$ detects a $\rightleftharpoons_0$-free subteam, refuting $\exists^1_\alpha \exists^1_\beta \chi_0(\alpha,\beta)$.

Intuitively, they quantify over subteams $S \subseteq T_\alpha$ (in case of $\exists^\subseteq_\alpha/\forall^\subseteq_\alpha$) or over worlds $w \in T_\alpha$ (for $\exists^1_\alpha/\forall^1_\alpha$), and require that the shrunk team T^α_S resp. T^α_w satisfies φ.

▶ **Proposition 4.3** (⋆). $\exists^\subseteq_\alpha, \forall^\subseteq_\alpha, \exists^1_\alpha, \forall^1_\alpha$ *have the following semantics:*

$$T \vDash \exists^\subseteq_\alpha \varphi \Leftrightarrow \exists S \subseteq T_\alpha \colon T^\alpha_S \vDash \varphi \qquad T \vDash \exists^1_\alpha \varphi \Leftrightarrow \exists w \in T_\alpha \colon T^\alpha_w \vDash \varphi$$
$$T \vDash \forall^\subseteq_\alpha \varphi \Leftrightarrow \forall S \subseteq T_\alpha \colon T^\alpha_S \vDash \varphi \qquad T \vDash \forall^1_\alpha \varphi \Leftrightarrow \forall w \in T_\alpha \colon T^\alpha_w \vDash \varphi$$

Proof sketch. Here, we sketch only the existential cases, as the universal ones work dually. The formula $\exists^\subseteq_\alpha \varphi := \alpha \vee \varphi$ allows to split T into subteams $U_1 \subseteq T_\alpha$ and U_2, where $U_2 \vDash \varphi$. As U_2 must contain $T_{\neg\alpha}$, clearly it is of the form T^α_S for some S. Conversely, every team of the form T^α_S induces a splitting of T into U_1, U_2 as above.

The singleton quantifier, $\exists^1_\alpha$, states that for some non-empty $U \subseteq T_\alpha$ it holds that $T^\alpha_S \vDash \varphi$ for every non-empty $S \subseteq U$. This is equivalent to $T^\alpha_U \vDash \varphi$ being true for some singleton $U \subseteq T_\alpha$. ◀

Implementing bisimulation

Finally, we have all ingredients to implement k-bisimulation in the following inductive manner:

$$\chi_0(\alpha,\beta) := (\alpha \vee \beta) \hookrightarrow \bigwedge_{p \in \Phi} {=}(p)$$

$$\chi_{k+1}(\alpha,\beta) := \chi_0(\alpha,\beta) \wedge \Box\chi^*_k(\alpha,\beta)$$

$$\chi^*_k(\alpha,\beta) := (\neg\alpha \wedge \neg\beta) \otimes \Big(\mathsf{E}\alpha \wedge \mathsf{E}\beta \wedge {\sim}\big[(\alpha \otimes \beta) \vee (\mathsf{E}\alpha \wedge \mathsf{E}\beta \wedge {\sim}\exists^1_\alpha\exists^1_\beta \chi_k(\alpha,\beta))\big]\Big)$$

Here, $\hookrightarrow$ is defined as on p. 9. Let us prove that these formulas define bisimulation:

▶ **Theorem 4.4** (⋆). *Let $k \geq 0$. For all Kripke structures $\mathcal{K}$, teams T in $\mathcal{K}$, disjoint scopes α, β in $\mathcal{K}$, and points $w \in T_\alpha$ and $v \in T_\beta$ it holds:*

$$T^{\alpha,\beta}_{w,v} \vDash \chi_k(\alpha,\beta) \text{ if and only if } w \rightleftharpoons_k v,$$
$$T \vDash \chi^*_k(\alpha,\beta) \text{ if and only if } T_\alpha \rightleftharpoons_k T_\beta.$$

Moreover, both $\chi_k(\alpha,\beta)$ and $\chi^*_k(\alpha,\beta)$ are MTL_k-formulas that are constructible in space $\mathcal{O}(\log(k + |\Phi| + |\alpha| + |\beta|))$.

Proof sketch. By induction on k. First, the formula $\chi_0(\alpha,\beta)$ expresses $w \rightleftharpoons_0 v$ when evaluated on a team $T^{\alpha,\beta}_{w,v}$. By the semantics of $\hookrightarrow$, $\chi_0(\alpha,\beta)$ is true if and only if $\{w,v\} \vDash {=}(p)$

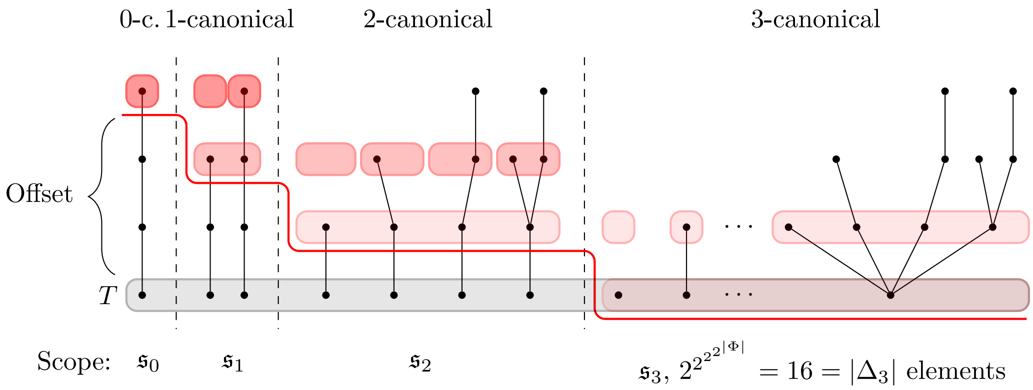

Figure 3 Visualization of the 3-staircase for $\Phi = \emptyset$, where the subteam $T_{\mathfrak{s}_i}$ is i-canonical with offset $3 - i$.

for all $p \in \Phi$. By definition of $=(\cdot)$, then $w \vDash p \Leftrightarrow v \vDash p$ for all $p \in \Phi$, i.e., $w \rightleftharpoons_0 v$. For χ_{k+1}, recall that $w \rightleftharpoons_{k+1} v$ is equivalent to $w \rightleftharpoons_0 v$ and $Rw \rightleftharpoons_k Rv$. Consequently, χ_{k+1} defines $(k+1)$-bisimilarity on points under the assumption that χ_k^* defines k-bisimilarity on teams.

Finally, $\chi_k^*(\alpha, \beta)$ checks $T_\alpha \rightleftharpoons_k T_\beta$ as follows. If at least one of these teams is empty, then it is easy to see that χ_k^* acts correctly. For non-empty T_α and T_β, the idea is to isolate any single point $z \in T_\alpha \cup T_\beta$ that serves as a *counter-example* against $[\![T_\alpha]\!]_k = [\![T_\beta]\!]_k$ by, say, $[\![z]\!]_k \in [\![T_\beta]\!]_k \setminus [\![T_\alpha]\!]_k$. We erase $T_\beta \setminus \{z\}$ from T using the disjunction $\vee$, as $T_\beta \setminus \{z\} \vDash \alpha \oslash \beta$. The remaining team is exactly T_z^β, in which $\exists^1_\alpha \exists^1_\beta \chi_k(\alpha, \beta)$ fails (see Figure 2). The case $[\![z]\!]_k \in [\![T_\alpha]\!]_k \setminus [\![T_\beta]\!]_k$ is detected analogously. Moreover, the formulas can be constructed in logspace in a straightforward manner, and $\mathsf{md}(\chi_k) = \mathsf{md}(\chi_k^*) = k$. ◀

Let us again stress that χ_k implements only an *approximation* of $\rightleftharpoons_k$, as it relies on scopes to be labeled in the structure correctly.

5 Enforcing a canonical model

As discussed before, we now aim at constructing an MTL_k-formula that is satisfiable but permits *only* k-canonical models. For $k = 0$, Hannula et al. [13] defined the PTL-formula

$$\mathsf{max}(X) := \sim \bigvee_{x \in X} =(x)$$

and proved that $T \vDash \mathsf{max}(\Phi)$ if and only if T is 0-canonical, i.e., contains all Boolean assignment over Φ. We generalize this for all k, i.e., construct a satisfiable formula canon_k that has only k-canonical models.

Staircase models

Our approach is to express k-canonicity by inductively enforcing i-canonical sets of worlds for $i = 0, \ldots, k$ located in different "height" inside the model. For this purpose, we employ distinct scopes $\mathfrak{s}_0, \ldots, \mathfrak{s}_k$ ("*stairs*"), and introduce a specific class of models:

▶ **Definition 5.1.** Let $k, i \geq 0$ and let $(\mathcal{K}, T)$ be a Kripke structure with team, $\mathcal{K} = (W, R, V)$. A team T is *k-canonical with offset i* if for every $\tau \in \Delta_k$ there exists $w \in T$ with $[\![R^i w]\!]_k = \{\tau\}$.

$(\mathcal{K}, T)$ is called *k-staircase* if for all $i \in \{0, \ldots, k\}$ we have that $T_{\mathfrak{s}_i}$ is i-canonical with offset $k - i$.

A 3-staircase for $\Phi = \emptyset$ is depicted in Figure 3, which is easily adapted for $\Phi \neq \emptyset$ and arbitrary k. In particular, it is a *directed forest*, which means that its underlying undirected graph is acyclic and all its worlds are either *roots* (i.e., without predecessor) or have exactly one predecessor. Moreover, it has bounded *height*, where the height of a directed forest is the greatest number h such that every path traverses at most h edges.

▶ **Proposition 5.2.** *For each $k \geq 0$, there is a finite k-staircase $(\mathcal{K}, T)$ such that $\mathfrak{s}_0, \ldots, \mathfrak{s}_k$ are disjoint scopes in $\mathcal{K}$, and $\mathcal{K}$ is a directed forest with height at most k and its set of roots being exactly T.*

Observe that a model being a k-staircase is a stronger condition than k-canonicity.

▶ **Corollary 5.3.** *Every satisfiable MTL_k-formula has a finite model $(\mathcal{K}, T)$ such that $\mathcal{K}$ is a directed forest with height at most k and its set of roots being exactly T.*

Enforcing canonicity

In the rest of the section, we illustrate how a k-staircase can be enforced in MTL inductively.

For $\Phi = \emptyset$, the inductive step – obtaining $(k+1)$-canonicity from k-canonicity – is captured by the formula $\forall^{\subseteq}_{\alpha} \exists^{1}_{\beta} \Box \chi^*_k(\alpha, \beta)$. It states that for every *subteam* $T' \subseteq T_\alpha$ there exists a *point* $w \in T_\beta$ such that $[\![RT']\!]_k = [\![Rw]\!]_k$. Intuitively, every possible set of types is captured as the image of some point in T_β. As a consequence, if T_α is k-canonical with offset 1, then T_β will be $(k+1)$-canonical.

Note that the straightforward formula $\Box^k \mathsf{max}(\Phi)$ expresses 0-canonicity of $R^k T$, but *not* 0-canonicity of T with offset k (consider, e.g., a singleton T). Instead, we use the formula

$$\mathsf{max\text{-}off}_i(\beta) := \beta \hookrightarrow \Big(\Diamond^i \top \wedge \big(\Box^i \mathsf{max}(\Phi)\big) \wedge \forall^1_\beta \Box^i \bigwedge_{p \in \Phi} {=}(p)\Big).$$

It states that $R^i T_\beta$ is 0-canonical, but that $R^i w$ admits only one propositional assignment for each $w \in T_\beta$. In this light, k-canonicity with offset i is altogether defined as follows:

$$\begin{aligned}
\rho^i_0(\beta) &:= \exists^{\subseteq}_{\beta} \, \mathsf{max\text{-}off}_i(\beta) \\
\rho^i_{k+1}(\alpha, \beta) &:= \forall^{\subseteq}_{\alpha} \exists^{\subseteq}_{\beta} \left(\rho^i_0(\beta) \wedge \Box^i \forall^1_\beta \Box \chi^*_k(\alpha, \beta)\right) \\
\mathsf{canon}_k &:= \rho^k_0(\mathfrak{s}_0) \wedge \bigwedge_{m=1}^{k} \rho^{k-m}_m(\mathfrak{s}_{m-1}, \mathfrak{s}_m)
\end{aligned}$$

▶ **Theorem 5.4** ($\star$)**.** *Let $k \geq 0$. The formula canon_k is an MTL_k-formula and constructible in space $\mathcal{O}(\log(|\Phi| + k))$.*

Moreover, if $\mathcal{K}$ is a Kripke structure with disjoint scopes $\mathfrak{s}_0, \ldots, \mathfrak{s}_k$, then $(\mathcal{K}, T) \vDash \mathsf{canon}_k$ if and only if $(\mathcal{K}, T)$ is a k-staircase.

Proof sketch. By induction on k. We sketch the induction step.

Suppose T_α is k-canonical with offset $i+1$. For each $S \subseteq T_\alpha$, the formula $\rho^i_{k+1}(\alpha, \beta)$ quantifies a subteam $U \subseteq T_\beta$ that is 0-canonical with offset i. Additionally, it also forces all points in $R^i U$ (and hence at least one point of every 0-type) to mimic the k-types of $R^{i+1} S$ in all points of their image. Together, this results in $(k+1)$-canonicity with offset i. ◀

It remains to demonstrate that the restriction of the $\mathfrak{s}_i$ being scopes *a priori* can be omitted, since we can, in a sense, define it in MTL as well. For this, let $\Psi \subseteq \mathcal{PS}$ be disjoint from Φ. Then the formula below ensures that Ψ is a set of disjoint scopes "up to height k", which is sufficient for our purposes.

$$\mathsf{scopes}_k(\Psi) := \bigwedge_{\substack{x,y\in\Psi \\ x\neq y}} \neg(x \wedge y) \wedge \bigwedge_{i=1}^{k} \Big((x \wedge \Box^i x) \vee (\neg x \wedge \Box^i \neg x)\Big).$$

▶ **Lemma 5.5.** *If $\varphi \in \mathsf{MTL}_k$, then φ is satisfiable if and only if $\varphi \wedge \Box^{k+1}\bot$ is satisfiable.*

Proof. As the direction from right to left is trivial, assume that φ is satisfiable. By Corollary 5.3, it then has a model $(\mathcal{K}, T)$ that is a directed forest of height at most k. But then $(\mathcal{K}, T) \vDash \Box^{k+1}\bot$, since $R^{k+1}T = \emptyset$ and $(\mathcal{K}, \emptyset)$ satisfies all ML-formulas, including $\bot$. ◀

▶ **Theorem 5.6.** $\mathsf{canon}_k \wedge \mathsf{scopes}_k(\{\mathfrak{s}_0, \dots, \mathfrak{s}_k\}) \wedge \Box^{k+1}\bot$ *is satisfiable, but has only k-staircases as models.*

Proof. By combining Proposition 5.2, Theorem 5.4 and Lemma 5.5, the formula is satisfiable. Since in every model $(\mathcal{K}, T)$ the propositions $\mathfrak{s}_0, \dots, \mathfrak{s}_k$ must be disjoint scopes due to $\Box^{k+1}$ and scopes_k, we can apply Theorem 5.4. ◀

Let us stress that the formula canon_k is again only an approximation of k-canonicity, since the scopes $\mathfrak{s}_0, \dots, \mathfrak{s}_{k-1}$ are necessary for the construction as well. However, both χ_k and canon_k being efficiently constructible is crucial for our main result in the next section.

6 Complexity lower bounds

In this section, we provide the matching lower bounds for Theorem 3.8 and Corollary 3.9:

▶ **Theorem 6.1.** $\mathsf{SAT}(\mathsf{MTL})$ *and* $\mathsf{VAL}(\mathsf{MTL})$ *are complete for* $\mathrm{TOWER}(\mathrm{poly})$. *For all $k \geq 0$,* $\mathsf{SAT}(\mathsf{MTL}_k)$ *and* $\mathsf{VAL}(\mathsf{MTL}_k)$ *are complete for* $\mathrm{ATIME\text{-}ALT}(\exp_{k+1}, \mathrm{poly})$.

The above complexity classes are complement-closed, and MTL and MTL_k are closed under negation. For this reason, it suffices to consider $\mathsf{SAT}(\mathsf{MTL})$ and $\mathsf{SAT}(\mathsf{MTL}_k)$. Moreover, the case $k = 0$ is equivalent to $\mathsf{SAT}(\mathsf{PTL})$ being $\mathrm{ATIME\text{-}ALT}(\exp, \mathrm{poly})$-hard, which was proven by Hannula et al. [14]. Their reduction works in logarithmic space.

Consequently, the result boils down to the following lemma:

▶ **Lemma 6.2.** *If $L \in \mathrm{TOWER}(\mathrm{poly})$, then $L \leq_{\mathrm{m}}^{\log} \mathsf{SAT}(\mathsf{MTL})$.*
If $k \geq 1$ and $L \in \mathrm{ATIME\text{-}ALT}(\exp_{k+1}, \mathrm{poly})$, then $L \leq_{\mathrm{m}}^{\log} \mathsf{SAT}(\mathsf{MTL}_k)$.

We devise for each L a reduction $x \mapsto \varphi_x$ such that φ_x is a formula that is satisfiable if and only if $x \in L$. By assumption, there exists a single-tape alternating Turing machine M that decides L (for $L \in \mathrm{TOWER}(\mathrm{poly})$, w.l.o.g. M is alternating as well). Then $M = (Q, \Gamma, \delta)$, where Q is the disjoint union of $Q_\exists$ (*existential states*), $Q_\forall$ (*universal states*), Q_{acc} (*accepting states*) and Q_{rej} (*rejecting states*). Also, Q contains some initial state q_0. Γ is the finite tape alphabet, $\flat$ the blank symbol, and δ the transition relation.

We design φ_x in a fashion that forces its models $(\mathcal{K}, T)$ to encode an accepting computation of M on x. Let us call any legal sequence of configurations of M (not necessarily starting with the initial configuration) a *run*. Then, similarly as in Cook's famous theorem [5], we encode runs as square "grids" with a vertical "time" coordinate and a horizontal "space" coordinate in the model, i.e., each row of the grid represents a configuration of M.

W.l.o.g. M has runtime at most N and tape cells $\{1, \dots, N\}$. A run of M is then a function $C\colon \{1, \dots, N\}^2 \to \Gamma \cup (Q \times \Gamma)$. In M's initial configuration, for instance, we have $C(1,1) = (q_0, x_1)$, $C(i,1) = x_i$ for $2 \leq i \leq n$, and $C(i,1) = \flat$ for $n < i \leq N$.

Due to the semantics of MTL, such a run must be encoded in $(\mathcal{K}, T)$ very carefully. We let T contain N^2 worlds $w_{i,j}$ in which the respective value of $C(i,j)$ is encoded in a propositional assignment. However, we cannot simply pursue the standard approach of assembling a large $N \times N$-grid in the edge relation R in order to compare successive configurations; by Corollary 5.3, we cannot force the model to contain R-paths longer than $|\varphi_x|$.

Instead, to define grid neighborship, we let $w_{i,j}$ encode i and j in its *type.* More precisely, we impose a linear order $\prec_k$ on Δ_k that is defined by an MTL_k-formula ζ_k. Then, instead of using $\Box$ and $\Diamond$, we examine the grid by letting ζ_k judge whether a given pair of worlds is deemed (horizontally or vertically) adjacent. Analogously to χ_k^*, we also define an order $\prec_k^*$ on teams via a formula ζ_k^*. Since order is a binary relation, the formulas are once more parameterized by two scopes:

$$\zeta_0(\alpha,\beta) := \bigvee_{p\in\Phi} \Big[(\alpha \hookrightarrow \neg p) \land (\beta \hookrightarrow p) \land \bigwedge_{\substack{q\in\Phi\\ q<p}} (\alpha \lor \beta) \hookrightarrow {=}(q)\Big]$$

$$\zeta_{k+1}(\alpha,\beta) := \zeta_0(\alpha,\beta) \ \varovee \ \big(\chi_0(\alpha,\beta) \land \Box\zeta_k^*(\alpha,\beta)\big)$$

$$\begin{aligned}\zeta_k^*(\alpha,\beta) := {} & \exists^1_{\mathfrak{s}_k}\ \big(\exists^1_\beta \chi_k(\mathfrak{s}_k,\beta)\big) \land \big({\sim}\exists^1_\alpha \chi_k(\mathfrak{s}_k,\alpha)\big)\\ & \land \Big(\big(\chi_k^*(\alpha,\beta) \land (\alpha \lor \beta)\big) \ \lor\ \big(\forall^1_{\alpha\lor\beta} {\sim}\zeta_k(\mathfrak{s}_k, \alpha\lor\beta)\big)\Big)\end{aligned}$$

We refer the reader to the full paper [30] for the proof that there exist orders $\prec_k$ and $\prec_k^*$ on Δ_k and $\mathfrak{P}(\Delta_k)$ that are defined by ζ_k and ζ_k^* in the following sense:

▶ **Theorem 6.3** ($\star$). *Let $k \geq 0$, and $(\mathcal{K}, T)$ be a k-staircase with disjoint scopes $\alpha, \beta, \mathfrak{s}_0, \dots, \mathfrak{s}_k$. If $w \in T_\alpha$ and $v \in T_\beta$, then*

$$T^{\alpha,\beta}_{w,v} \vDash \zeta_k(\alpha,\beta) \quad \textit{if and only if} \quad [\![w]\!]_k \prec_k [\![v]\!]_k,$$
$$T \vDash \zeta_k^*(\alpha,\beta) \quad \textit{if and only if} \quad [\![T_\alpha]\!]_k \prec_k^* [\![T_\beta]\!]_k.$$

Furthermore, both $\zeta_k(\alpha,\beta)$ and $\zeta_k^*(\alpha,\beta)$ are MTL_k-formulas that are constructible in space $\mathcal{O}(\log(k + |\Phi| + |\alpha| + |\beta|))$.

Encoding runs in a team

Next, we discuss in more detail how runs $C\colon \{1, \dots, N\}^2 \to \Gamma \cup (Q \times \Gamma)$ are encoded in a team T. Given a world $w \in T$, we partition the image Rw with two special propositions $\mathfrak{t} \notin \Phi$ ("timestep") and $\mathfrak{p} \notin \Phi$ ("position"). Then we assign to w the pair $\ell(w) := (i,j)$ such that $[\![(Rw)_\mathfrak{t}]\!]_{k-1}$ is the i-th element, and $[\![(Rw)_\mathfrak{p}]\!]_{k-1}$ is the j-th element in the order $\prec_{k-1}^*$. We call the pair $\ell(w)$ the *location* of w (in the grid).

Accordingly, we fix $N := |\mathfrak{P}(\Delta_{k-1}^\Phi)|$. For the case of fixed k, M has runtime bounded by $\exp_{k+1}(g(n))$ for a polynomial g. Then taking $\Phi := \{p_1, \dots, p_{g(n)}\}$ yields a sufficiently large coordinate space, as

$$\exp_{k+1}(g(n)) = \exp_{k+1}(|\Phi|) = 2^{\exp_{k-1}\left(2^{|\Phi|}\right)} \leq 2^{\exp^*_{k-1}\left(2^{|\Phi|}\right)} = 2^{|\Delta_{k-1}^\Phi|} = |\mathfrak{P}(\Delta_{k-1}^\Phi)|$$

by Proposition 3.5. Likewise, if in the second case M has runtime bounded by $\exp_{g(n)}(1)$, we let $\Phi := \emptyset$ and compute $k := g(|x|) + 1$, but otherwise proceed identically.

Next, let Ξ be a constant set of propositions disjoint from Φ that encodes the range of C via some bijection $c\colon \Xi \to \Gamma \cup (Q \times \Gamma)$. If a world w satisfies exactly one proposition p of those in Ξ, then we define $c(w) := c(p)$. Intuitively, $c(w)$ is the *content* of the grid cell represented by w.

Using ℓ and c, the function C can be encoded into a team T as follows. First, a team T is called *grid* if every point in T satisfies exactly one proposition in Ξ, and if every location $(i, j) \in \{1, \ldots, N\}^2$ occurs as $\ell(w)$ for some point $w \in T$. Moreover, a grid T is called *pre-tableau* if for every location (i, j) *and* every element $p \in \Xi$ there is some world $w \in T$ such that $\ell(w) = (i, j)$ and $w \vDash p$. Finally, a grid T is a *tableau* if any two elements $w, w' \in T$ with $\ell(w) = \ell(w')$ also agree on Ξ, i.e., $c(w) = c(w')$.

Let us motivate the above definitions. Clearly, the definition of a grid T means that T captures the whole domain of C, and that c is well-defined on the level of *points*. If T is additionally a tableau, then c is also well-defined on the level of *locations*. In other words, every tableau T induces a function $C_T\colon \{1, \ldots, N\}^2 \to \Gamma \cup (Q \times \Gamma)$ via $C_T(i, j) := c(w)$, where $w \in T$ is arbitrary such that $\ell(w) = (i, j)$. Finally, a pre-tableau is, roughly speaking, the "union" of all possible C. In particular, given any pre-tableau, the definition ensures that arbitrary tableaus can be obtained from it by the means of subteam quantification $\exists^{\subseteq}$ (cf. p. 9).

A tableau T is *legal* if C_T is a run of M, i.e., if every row is a configuration of M, and if every pair of two successive rows represents a valid δ-transition.

The idea of the reduction is now to capture the alternating computation of M by nesting polynomially many quantifications (via $\exists^{\subseteq}$ and $\forall^{\subseteq}$) of legal tableaus, of which each one is the continuation of the computation of the previous one. For this purpose, we devise formulas such as $\psi_{\text{pre-tableau}}(\alpha)$ and $\psi_{\text{legal}}(\alpha)$ that express that T_α is a pre-tableau, or a legal tableau, respectively. These formulas rely on canon_k to achieve a sufficiently large team, and on ζ_k resp. ζ_k^* for accessing adjacent grid cells in order to verify the transitions between configurations.

Due to space constraints, we cannot present their implementation here. Instead, we refer the reader to the appendix or the full version of the paper [30] for details.

7 Concluding remarks

In Theorem 6.1, we settled the open question of the complexity of MTL and established TOWER(poly)-completeness for its satisfiability and validity problem. Likewise, the fragments MTL_k are proved complete for ATIME-ALT$(\exp_{k+1}, \text{poly})$, the levels of the elementary hierarchy with polynomially many alternations.

As our main tool, we introduced a suitable notion of canonical models for modal logics with team semantics. We showed that such models exist for MTL and MTL_k, and that some satisfiable MTL_k-formulas of polynomial size have *only* k-canonical models.

Our lower bounds carry over to two-variable first-order team logic $\mathsf{FO}^2(\sim)$ and its fragment $\mathsf{FO}^2_k(\sim)$ of bounded quantifier rank k as well [29]. While the former is TOWER(poly)-complete, the latter is ATIME-ALT$(\exp_{k+1}, \text{poly})$-hard. However, no matching upper bound for the satisfiability problem of $\mathsf{FO}^2_k(\sim)$ exists.

In future research, it could be useful to further generalize the concept of canonical models for other logics with team semantics. Do logics such as $\mathsf{FO}^2_k(\sim)$ permit a canonical model in the spirit of k-canonical models for MTL_k, and does this yield a tight upper bound on the complexity of their satisfiability problem? How do MTL_k and $\mathsf{FO}^2_k(\sim)$ differ in terms of succinctness?

References

1 Samson Abramsky, Juha Kontinen, Jouko Väänänen, and Heribert Vollmer, editors. *Dependence Logic, Theory and Applications.* Springer, 2016. doi:10.1007/978-3-319-31803-5.

2 Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal logic.* Cambridge University Press, New York, NY, USA, 2001.

3 Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, 1981. doi:10.1145/322234.322243.

4 Kevin J. Compton and C. Ward Henson. A Uniform Method for Proving Lower Bounds on the Computational Complexity of Logical Theories. *Ann. Pure Appl. Logic*, 48(1):1–79, 1990. doi:10.1016/0168-0072(90)90080-L.

5 Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971. doi:10.1145/800157.805047.

6 Arnaud Durand, Juha Kontinen, and Heribert Vollmer. *Expressivity and Complexity of Dependence Logic*, pages 5–32. Springer International Publishing, 2016. doi:10.1007/978-3-319-31803-5_2.

7 Pietro Galliani. Inclusion and exclusion dependencies in team semantics - on some logics of imperfect information. *Ann. Pure Appl. Logic*, 163(1):68–84, 2012. doi:10.1016/j.apal.2011.08.005.

8 Pietro Galliani. Upwards closed dependencies in team semantics. *Information and Computation*, 245:124–135, 2015. doi:10.1016/j.ic.2015.06.008.

9 Erich Grädel and Jouko Väänänen. Dependence and independence. *Studia Logica*, 101(2):399–410, 2013. doi:10.1007/s11225-013-9479-2.

10 Miika Hannula. Validity and entailment in modal and propositional dependence logics. *CoRR*, abs/1608.04301, 2016. URL: http://arxiv.org/abs/1608.04301.

11 Miika Hannula. Validity and Entailment in Modal and Propositional Dependence Logics. In *26th EACSL Annual Conference on Computer Science Logic (CSL 2017)*, volume 82 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:17. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.CSL.2017.28.

12 Miika Hannula, Juha Kontinen, Martin Lück, and Jonni Virtema. On Quantified Propositional Logics and the Exponential Time Hierarchy. In *Proceedings of the Seventh International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2016*, pages 198–212, 2016. doi:10.4204/EPTCS.226.14.

13 Miika Hannula, Juha Kontinen, Jonni Virtema, and Heribert Vollmer. Complexity of propositional independence and inclusion logic. In *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015*, pages 269–280, 2015. doi:10.1007/978-3-662-48057-1_21.

14 Miika Hannula, Juha Kontinen, Jonni Virtema, and Heribert Vollmer. Complexity of Propositional Logics in Team Semantic. *ACM Transactions on Computational Logic*, 19(1):1–14, jan 2018. doi:10.1145/3157054.

15 Lauri Hella, Antti Kuusisto, Arne Meier, and Jonni Virtema. Model checking and validity in propositional and modal inclusion logics. In *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017*, pages 32:1–32:14, 2017. doi:10.4230/LIPIcs.MFCS.2017.32.

16 Lauri Hella, Antti Kuusisto, Arne Meier, and Heribert Vollmer. Modal Inclusion Logic: Being Lax is Simpler than Being Strict. In *Mathematical Foundations of Computer Science 2015*, volume 9234, pages 281–292. Springer Berlin Heidelberg, 2015. URL: http://link.springer.com/10.1007/978-3-662-48057-1_22.

17 Lauri Hella, Kerkko Luosto, Katsuhiko Sano, and Jonni Virtema. The expressive power of modal dependence logic. In *Advances in Modal Logic 10, invited and contributed papers*

from the tenth conference on "Advances in Modal Logic". Groningen, The Netherlands, August 5-8, 2014, pages 294–312, 2014. URL: `http://www.aiml.net/volumes/volume10/Hella-Luosto-Sano-Virtema.pdf`.

18 Lauri Hella and Johanna Stumpf. The expressive power of modal logic with inclusion atoms. *Electronic Proceedings in Theoretical Computer Science*, 193:129–143, 2015. `doi:10.4204/EPTCS.193.10`.

19 Jaakko Hintikka and Gabriel Sandu. Informational Independence as a Semantical Phenomenon. In Jens Erik Fenstad, Ivan T. Frolov, and Risto Hilpinen, editors, *Logic, Methodology and Philosophy of Science VIII*, volume 126 of *Studies in Logic and the Foundations of Mathematics*, pages 571–589. Elsevier, 1989. `doi:10.1016/S0049-237X(08)70066-1`.

20 Wilfrid Hodges. Compositional semantics for a language of imperfect information. *Logic Journal of IGPL*, 5(4):539–563, 1997. `doi:10.1093/jigpal/5.4.539`.

21 M. Jonáš and J. Strejček. On the complexity of the quantified bit-vector arithmetic with binary encoding. *Information Processing Letters*, 2018. `doi:10.1016/j.ipl.2018.02.018`.

22 Juha Kontinen, Julian-Steffen Müller, Henning Schnoor, and Heribert Vollmer. A Van Benthem theorem for modal team semantics. In *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany*, pages 277–291, 2015. `doi:10.4230/LIPIcs.CSL.2015.277`.

23 Juha Kontinen, Julian-Steffen Müller, Henning Schnoor, and Heribert Vollmer. Modal independence logic. *Journal of Logic and Computation*, 27(5):1333–1352, 2017. `doi:10.1093/logcom/exw019`.

24 Juha Kontinen and Ville Nurmi. Team logic and second-order logic. *Fundam. Inform.*, 106(2-4):259–272, 2011. `doi:10.3233/FI-2011-386`.

25 Andreas Krebs, Arne Meier, and Jonni Virtema. A team based variant of CTL. In *22nd International Symposium on Temporal Representation and Reasoning, TIME 2015*, pages 140–149, 2015. `doi:10.1109/TIME.2015.11`.

26 Peter Lohmann and Heribert Vollmer. Complexity results for modal dependence logic. *Studia Logica*, 101(2):343–366, 04 2013. `doi:10.1007/s11225-013-9483-6`.

27 Martin Lück. Axiomatizations for propositional and modal team logic. In *25th EACSL Annual Conference on Computer Science Logic, CSL 2016*, pages 33:1–33:18, 2016. `doi:10.4230/LIPIcs.CSL.2016.33`.

28 Martin Lück. The power of the filtration technique for modal logics with team semantics. In *26th EACSL Annual Conference on Computer Science Logic, CSL 2017*, pages 31:1–31:20, 2017. `doi:10.4230/LIPIcs.CSL.2017.31`.

29 Martin Lück. On the complexity of team logic and its two-variable fragment. MFCS 2018. To appear.

30 Martin Lück. Canonical models and the complexity of modal team logic. *CoRR*, abs/1709.05253, 2017. URL: `https://arxiv.org/abs/1709.05253`.

31 Julian-Steffen Müller. *Satisfiability and model checking in team based logics*. PhD thesis, University of Hanover, 2014. URL: `http://d-nb.info/1054741921`.

32 Sylvain Schmitz. Complexity hierarchies beyond elementary. *TOCT*, 8(1):3:1–3:36, 2016. `doi:10.1145/2858784`.

33 Merlijn Sevenster. Model-theoretic and computational properties of modal dependence logic. *J. Log. Comput.*, 19(6):1157–1173, 2009. `doi:10.1093/logcom/exn102`.

34 Larry J. Stockmeyer and Albert R. Meyer. Word Problems Requiring Exponential Time: Preliminary Report. In *Proceedings of the 5th Annual ACM Symposium on Theory of Computing*, pages 1–9, 1973. `doi:10.1145/800125.804029`.

35 Jouko Väänänen. Modal dependence logic. *New perspectives on games and interaction*, 4:237–254, 2008.

36 Jonni Virtema. Complexity of validity for propositional dependence logics. *Inf. Comput.*, 253:224–236, 2017. doi:10.1016/j.ic.2016.07.008.

37 Marco Voigt. A fine-grained hierarchy of hard problems in the separated fragment. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017*, pages 1–12, 2017. doi:10.1109/LICS.2017.8005094.

38 Jouko Väänänen. *Dependence logic: A New Approach to Independence Friendly Logic.* Number 70 in London Mathematical Society student texts. Cambridge University Press, Cambridge ; New York, 2007.

39 Fan Yang and Jouko Väänänen. Propositional logics of dependence. *Ann. Pure Appl. Logic*, 167(7):557–589, 2016. doi:10.1016/j.apal.2016.03.003.

40 Fan Yang and Jouko Väänänen. Propositional team logics. *Ann. Pure Appl. Logic*, 168(7):1406–1441, 2017. doi:10.1016/j.apal.2017.01.007.

A Details of the reduction (Lemma 6.2)

In the appendix, we present our lower bound in detail:

▶ **Lemma 6.2.** *If* $L \in \mathrm{TOWER}(\mathrm{poly})$*, then* $L \leq^{\log}_{\mathrm{m}} \mathsf{SAT}(\mathsf{MTL})$*.*
If $k > 0$ *and* $L \in \mathrm{ATIME\text{-}ALT}(\exp_{k+1}, \mathrm{poly})$*, then* $L \leq^{\log}_{\mathrm{m}} \mathsf{SAT}(\mathsf{MTL}_k)$*.*

We describe the reduction $x \mapsto \varphi_x$. In what follows, let $n := |x|$. The correctness proof for the reduction will be built on several claims. These claims are not hard to derive, and for detailed proofs of all steps we refer the reader to the full version of the paper [30].

An discussed in Section 6, we choose to represent a location (i, j) in a point w as a pair (Δ', Δ'') by stipulating that $\Delta' = [\![(Rw)_{\mathfrak{t}}]\!]_{k-1}$ and $\Delta'' = [\![(Rw)_{\mathfrak{p}}]\!]_{k-1}$, where $\mathfrak{t}$ ("time") and $\mathfrak{p}$ ("position") are special propositions in $\mathcal{PS} \setminus \Phi$. To access the two components of a encoded location independently, we introduce the operator $|^{\alpha}_{\mathfrak{q}}\, \psi := (\alpha \wedge \neg\mathfrak{q}) \vee ((\alpha \hookrightarrow \mathfrak{q}) \wedge \psi)$, where $\mathfrak{q} \in \{\mathfrak{t}, \mathfrak{p}\}$ and $\alpha \in \mathsf{ML}$. It is easy to check that $T \vDash |^{\alpha}_{\mathfrak{q}}\, \psi$ iff $T^{\alpha}_{T_{\mathfrak{q}}} \vDash \psi$.

In order to *compare* the locations of grid cells, for $\mathfrak{q} \in \{\mathfrak{t}, \mathfrak{p}\}$ we define the formulas $\psi^{\mathfrak{q}}_{\prec}(\alpha, \beta)$, which tests whether the location in T_α is less than the one in T_β w.r.t. its $\mathfrak{q}$-component (assuming singleton teams T_α and T_β), and $\psi^{\mathfrak{q}}_{\equiv}(\alpha, \beta)$ which checks for equality of the respective component:

$$\psi^{\mathfrak{q}}_{\prec}(\alpha, \beta) := \Box\, |^{\alpha}_{\mathfrak{q}}|^{\beta}_{\mathfrak{q}}\, \zeta^{*}_{k-1}(\alpha, \beta) \qquad\qquad \psi^{\mathfrak{q}}_{\equiv}(\alpha, \beta) := \Box\, |^{\alpha}_{\mathfrak{q}}|^{\beta}_{\mathfrak{q}}\, \chi^{*}_{k-1}(\alpha, \beta)$$

For this purpose, $\psi^{\mathfrak{q}}_{\prec}$ is built upon the formula ζ^{*}_{k-1} from Theorem 6.3, while $\psi^{\mathfrak{q}}_{\equiv}$ checks for equality with the help of χ^{*}_{k-1} from Theorem 4.4.

▶ **Claim** (a). *Let* $\mathcal{K}$ *be a structure with a team* T *and disjoint scopes* α *and* β*.*
Suppose $w \in T_\alpha$ *and* $v \in T_\beta$*, where* $\ell(w) = (i_w, j_w)$ *and* $\ell(v) = (i_v, j_v)$*. Then:*

$$T^{\alpha,\beta}_{w,v} \vDash \psi^{\mathfrak{t}}_{\equiv}(\alpha, \beta) \;\Leftrightarrow\; i_w = i_v \qquad\qquad T^{\alpha,\beta}_{w,v} \vDash \psi^{\mathfrak{p}}_{\equiv}(\alpha, \beta) \;\Leftrightarrow\; j_w = j_v.$$

Moreover, if $\alpha, \beta, \mathfrak{s}_0, \ldots, \mathfrak{s}_k$ *are disjoint scopes in* $\mathcal{K}$ *and* $(\mathcal{K}, T)$ *is a* k*-staircase, then:*

$$T^{\alpha,\beta}_{w,v} \vDash \psi^{\mathfrak{t}}_{\prec}(\alpha, \beta) \;\Leftrightarrow\; i_w < i_v \qquad\qquad T^{\alpha,\beta}_{w,v} \vDash \psi^{\mathfrak{p}}_{\prec}(\alpha, \beta) \;\Leftrightarrow\; j_w < j_v.$$

Next, we construct formulas that check whether a given team is a grid, pre-tableau, or a tableau, respectively. To check that every location $(i, j) \in \{1, \ldots, N\}^2$ of the grid occurs as $\ell(w)$ of some $w \in T$, we quantify over all pairs $(\Delta', \Delta'') \in \mathfrak{P}(\Delta_{k-1})^2$. To cover all these sets of types we can quantify, for instance, over the images of all points of $T_{\mathfrak{s}_k}$. As we cannot pick *two* subteams from the same scope at once, we enforce a k-canonical copy $T_{\mathfrak{s}'_k}$ of $T_{\mathfrak{s}_k}$ in the spirit of Theorem 5.4:

$$\mathsf{canon}' := \rho_0^k(\mathfrak{s}_0) \wedge \bigwedge_{m=1}^{k} \rho_m^{k-m}(\mathfrak{s}_{m-1}, \mathfrak{s}_m) \wedge \rho_k^0(\mathfrak{s}_{k-1}, \mathfrak{s}'_k)$$

▶ **Claim (b).** *If $\mathfrak{s}_0, \ldots, \mathfrak{s}_k, \mathfrak{s}'_k$ are disjoint scopes in $\mathcal{K}$, then $(\mathcal{K}, T) \vDash \mathsf{canon}'$ if and only if $(\mathcal{K}, T)$ is a k-staircase and $T_{\mathfrak{s}'_k}$ is k-canonical.*

Moreover, $\mathsf{canon}' \wedge \mathsf{scopes}_k(\{\mathfrak{s}_0, \ldots, \mathfrak{s}_k, \mathfrak{s}'_k\}) \wedge \Box^{k+1}\bot$ is satisfiable, but is only satisfied by k-staircases $(\mathcal{K}, T)$ in which both $T_{\mathfrak{s}_k}$ and $T_{\mathfrak{s}'_k}$ are k-canonical. Furthermore, both formulas are constructible in space $\mathcal{O}(\log(|\Phi| + k))$.

The next formulas define grids resp. pre-tableaus.

$$\psi_{\text{pair}}(\alpha) := \Box\left[\left(|^{\alpha}_{\mathfrak{t}}\ \chi^*_{k-1}(\mathfrak{s}_k, \alpha)\right) \wedge \left(|^{\alpha}_{\mathfrak{p}}\ \chi^*_{k-1}(\mathfrak{s}'_k, \alpha)\right)\right]$$

$$\psi_{\text{grid}}(\alpha) := \Big(\alpha \hookrightarrow \bigvee_{e \in \Xi} e \wedge \bigwedge_{\substack{e' \in \Xi \\ e' \neq e}} \neg e'\Big) \wedge \forall^1_{\mathfrak{s}_k} \forall^1_{\mathfrak{s}'_k} \exists^1_{\alpha}\, \psi_{\text{pair}}(\alpha)$$

$$\psi_{\text{pre-tableau}}(\alpha) := \psi_{\text{grid}}(\alpha) \wedge \forall^1_{\mathfrak{s}_k} \forall^1_{\mathfrak{s}'_k} \bigwedge_{e \in \Xi} \exists^1_{\alpha}\big(\psi_{\text{pair}}(\alpha) \wedge (\alpha \hookrightarrow e)\big)$$

In all subsequent claims, we always assume that T is a team in a Kripke structure $\mathcal{K}$ such that $(\mathcal{K}, T)$ satisfies $\mathsf{canon}' \wedge \Box^{k+1}\bot$. Moreover, all stated scopes are always assumed pairwise disjoint in $\mathcal{K}$ (as we can enforce this later in the reduction with $\mathsf{scopes}_k(\cdots)$).

▶ **Claim (c).** *$T \vDash \psi_{grid}(\alpha)$ if and only if T_α is a grid and $T \vDash \psi_{pre\text{-}tableau}(\alpha)$ if and only if T_α is a pre-tableau.*

The other special case of a grid, that is, a *tableau*, requires a more elaborate approach to define in MTL. The difference to a grid or pre-tableau is that we have to quantify over all *pairs* (w, w') of points in T, and check that they agree on Ξ if $\ell(w) = \ell(w')$. However, as discussed before, while $\forall^1$ can quantify over all points in a team, it cannot quantify over pairs. As a workaround, we consider not only a tableau T_α, but also a *second* tableau that acts as a copy of T_α. Formally, for grids T_α, T_β, let $T_\alpha \approx T_\beta$ denote that for all pairs $(w, w') \in T_\alpha \times T_\beta$ it holds that $\ell(w) = \ell(w')$ implies $c(w) = c(w')$.

As $\approx$ is symmetric and transitive, $T_\alpha \approx T_\beta$ in fact implies both $T_\alpha \approx T_\alpha$ and $T_\beta \approx T_\beta$, and hence that both T_α and T_β are tableaus such that $C_{T_\alpha} = C_{T_\beta}$, where $C_{T_\alpha}, C_{T_\beta} \colon \{1, \ldots, N\}^2 \to \Gamma \cup (Q \times \Gamma)$ are the induced runs as discussed on p. 15.

$$\psi_{\text{tableau}}(\alpha) := \psi_{\text{grid}}(\alpha) \wedge \exists^{\subseteq}_{\gamma_0}\, \psi_{\text{grid}}(\gamma_0) \wedge \psi_{\approx}(\alpha, \gamma_0)$$

$$\psi_{\approx}(\alpha, \beta) := \forall^1_\alpha \forall^1_\beta \left(\left(\psi^{\mathfrak{t}}_{\equiv}(\alpha, \beta) \wedge \psi^{\mathfrak{p}}_{\equiv}(\alpha, \beta)\right) \twoheadrightarrow \bigovee_{e \in \Xi}((\alpha \vee \beta) \hookrightarrow e)\right)$$

In the following claim (and in the subsequent ones), we use the scopes $\gamma_0, \gamma_1, \gamma_2, \ldots$ as "auxiliary pre-tableaus". Later, we will also use them as domains to quantify extra locations or rows from. (The index of γ_i is incremented whenever necessary to avoid quantifying from the same scope twice.) For this reason, from now on we always assume, for sufficiently large i, that T_{γ_i} is a pre-tableau. This can be later enforced in the reduction with $\psi_{\text{pre-tableau}}(\gamma_i)$.

▶ **Claim (d).** *$T \vDash \psi_{tableau}(\alpha)$ if and only if T_α is a tableau.*

For grids T_α, T_β, it holds $T \vDash \psi_{\approx}(\alpha, \beta)$ if and only if $T_\alpha \approx T_\beta$.

To ascertain that a tableau contains a run of M, we have to check whether each row indeed is a configuration of M and whether consecutive configurations adhere to the transition relation δ of M. For the latter, in the spirit of Cook's theorem [5], it suffices to consider all *legal windows* in the grid, i.e., cells that are adjacent as follows, where $e_1, \ldots, e_6 \in \Gamma \cup (Q \times \Gamma)$:

e_1	e_2	e_3
e_4	e_5	e_6

If, say, $(q, a, q', a', R) \in Q \times \Gamma \times Q \times \Gamma \times \{L, R, N\}$ is a transition – M switches to state q' from q, replacing a on the tape by a', and moves to the right – then the windows obtained by setting $e_1 = e_4 = b$, $e_2 = (q, a)$, $e_5 = a'$, $e_3 = b'$, $e_6 = (q', b')$ are legal for all $b, b' \in \Gamma$. Using this scheme, δ is completely represented by some constant finite set $\mathsf{win} \subseteq \Xi^6$ of tuples $(e_1, \ldots, e_6)$ that represent the allowed windows in a run of M.

Let us next explain how adjacency of cells is expressed. Suppose that two points $w \in T_\alpha$ and $v \in T_\beta$ are given. That v is the immediate ($\mathfrak{t}$- or $\mathfrak{p}$-)successor of w then means that no element of the order exists between them. Simultaneously, w and v have to agree on the other component of their location, which is expressed by the first conjunct below. Formally, if $\mathfrak{q} \in \{\mathfrak{t}, \mathfrak{p}\}$ and $\overline{\mathfrak{q}} \in \{\mathfrak{t}, \mathfrak{p}\} \setminus \{\mathfrak{q}\}$, then we define:

$$\psi^{\mathfrak{q}}_{\text{succ}}(\alpha, \beta) := \psi^{\overline{\mathfrak{q}}}_{\equiv}(\alpha, \beta) \land \psi^{\mathfrak{q}}_{\prec}(\alpha, \beta) \land {\sim}\exists^1_{\gamma_0} \left(\psi^{\mathfrak{q}}_{\prec}(\alpha, \gamma_0) \land \psi^{\mathfrak{q}}_{\prec}(\gamma_0, \beta)\right)$$

▶ **Claim** (e). *If $w \in T_\alpha$ and $v \in T_\beta$, then:*

$$T^{\alpha,\beta}_{w,v} \vDash \psi^{\mathfrak{t}}_{succ}(\alpha, \beta) \Leftrightarrow \exists i, j \in \{1, \ldots, N\} \colon \ell(w) = (i, j) \text{ and } \ell(v) = (i+1, j)$$
$$T^{\alpha,\beta}_{w,v} \vDash \psi^{\mathfrak{p}}_{succ}(\alpha, \beta) \Leftrightarrow \exists i, j \in \{1, \ldots, N\} \colon \ell(w) = (i, j) \text{ and } \ell(v) = (i, j+1)$$

In this vein, we proceed by quantifying windows in the tableau T_α by quantifying elements from *six* tableaus $T_{\gamma_1}, \ldots, T_{\gamma_6}$ that are copies of T_α. For this purpose, we abbreviate

$$\exists^{\approx\alpha}_{\gamma_i} \varphi := \exists^{\subseteq}_{\gamma_i} \psi_{\text{grid}}(\gamma_i) \land \psi_{\approx}(\alpha, \gamma_i) \land \varphi.$$

Intuitively, under the premise that T_{γ_i} is a pre-tableau and T_α is a tableau, it "copies the tableau T_α into T_{γ_i}" by shrinking T_{γ_i} accordingly. This is proven analogously to Claim (d). The next formula states that the picked points are adjacent as shown in the picture below:

$$\psi_{\text{window}}(\gamma_1, \ldots, \gamma_6) := \bigwedge_{i \in \{1,2,3\}} \psi^{\mathfrak{t}}_{\text{succ}}(\gamma_i, \gamma_{i+3}) \land \psi^{\mathfrak{p}}_{\text{succ}}(\gamma_1, \gamma_2) \land \psi^{\mathfrak{p}}_{\text{succ}}(\gamma_2, \gamma_3)$$

Based on the above two, the formula defining legal tableaus follows.

$$\psi_{\text{legal}}(\alpha) := \psi_{\text{tableau}}(\alpha) \land \exists^{\approx\alpha}_{\gamma_1} \cdots \exists^{\approx\alpha}_{\gamma_6} \vartheta_1 \land \vartheta_2 \land \vartheta_3$$

We check that no two distinct cells in any row both contain a state of M:

$$\vartheta_1 := \forall^1_{\gamma_1} \forall^1_{\gamma_2} \left(\psi^{\mathfrak{t}}_{\equiv}(\gamma_1, \gamma_2) \land \psi^{\mathfrak{p}}_{\prec}(\gamma_1, \gamma_2)\right) \twoheadrightarrow \bigwedge_{(q_1,a_1),(q_2,a_q) \in Q \times \Gamma} {\sim}\left((\gamma_1 \hookrightarrow c^{-1}(q_1, a_1)) \land (\gamma_2 \hookrightarrow c^{-1}(q_2, a_2))\right)$$

We also check that every row contains a state. Intuitively, $\forall^1_{\gamma_1}$ fixes some row and $\exists^1_{\gamma_2} \psi^{\mathfrak{t}}_{\equiv}(\gamma_1, \gamma_2)$ searches that particular row for a state:

$$\vartheta_2 := \forall^1_{\gamma_1} \exists^1_{\gamma_2} \psi^{\mathfrak{t}}_{\equiv}(\gamma_1, \gamma_2) \land \bigovee_{(q,a) \in Q \times \Gamma} (\gamma_2 \hookrightarrow c^{-1}(q, a))$$

Finally, every window must be valid:

$$\vartheta_3 := \forall^1_{\gamma_1} \cdots \forall^1_{\gamma_6} \left(\psi_{\text{window}}(\gamma_1, \dots, \gamma_6) \rightarrowtriangle \bigovee_{(e_1,\dots,e_6)\in\text{win}} \bigwedge_{i=1}^{6} (\gamma_i \hookrightarrow e_i)\right)$$

▶ **Claim (f).** *$T \vDash \psi_{legal}(\alpha)$ iff T_α is a legal tableau, i.e., C_{T_α} exists and is a run of M.*

To now encode the initial configuration on input $x = x_1 \cdots x_n$ in a tableau, we access the first n cells of the first row and assign the respective letter of x, as well as the initial state to the first cell. Moreover, we assign $\flat$ to all other cells in that row. For each $\mathfrak{q} \in \{\mathfrak{t}, \mathfrak{p}\}$, we can check whether the location of a point in T_α is minimal in its $\mathfrak{q}$-component:

$$\psi^{\mathfrak{q}}_{\text{min}}(\alpha) := {\sim}\exists^1_{\gamma_0} \psi^{\mathfrak{q}}_{\prec}(\gamma_0, \alpha)$$

This enables us to fix the first row of the configuration:

$$\begin{aligned}\psi_{\text{input}}(\alpha) := \exists^{\approx\alpha}_{\gamma_1} \cdots \exists^{\approx\alpha}_{\gamma_{n+1}} \exists^1_{\gamma_1} \cdots \exists^1_{\gamma_n} \psi^{\mathfrak{t}}_{\text{min}}(\gamma_1) \wedge \psi^{\mathfrak{p}}_{\text{min}}(\gamma_1) \wedge \left(\gamma_1 \hookrightarrow c^{-1}(q_0, x_1)\right) \\ \bigwedge_{i=2}^{n} \psi^{\mathfrak{p}}_{\text{succ}}(\gamma_{i-1}, \gamma_i) \ \wedge \left(\gamma_i \hookrightarrow c^{-1}(x_i)\right) \\ \wedge \forall^1_{\gamma_{n+1}} \Big(\big(\psi^{\mathfrak{t}}_{\equiv}(\gamma_n, \gamma_{n+1})\big) \wedge \psi^{\mathfrak{p}}_{\prec}(\gamma_n, \gamma_{n+1})\big) \rightarrowtriangle \left(\gamma_{n+1} \hookrightarrow c^{-1}(\flat)\right)\Big)\end{aligned}$$

▶ **Claim (g).** *Let T_α be a tableau. Then $T \vDash \psi_{input}(\alpha)$ if and only if $C_{T_\alpha}(1,1) = (q_0, x_1)$, $C_{T_\alpha}(1,i) = x_i$ for $2 \le i \le n$, and $C_{T_\alpha}(1,i) = \flat$ for $n < i \le N$.*

Until now, we ignored the fact that M alternates between universal and existential branching polynomially often. To simulate this, we quantify polynomially many tableaus in an alternating fashion, each containing a part of the computation of M.

Each of these tableaus should possess a *tail configuration*, which is the configuration where M either accepts, rejects, or alternates from existential to universal branching or vice versa. Formally, a number $i \in \{1, \dots, N\}$ is a *tail index* of C if there exists j such that either

1. $C(i,j)$ has an accepting or rejecting state,
2. or $C(i,j)$ has an existential state and and there are $i' < i$ and j' with a universal state in $C(i',j')$,
3. or $C(i,j)$ has a universal state and there are $i' < i$ and j' with an existential state in $C(i',j')$.

The least such i is called *first tail index*, and the corresponding configuration is the *first tail configuration.*

The idea is that we can split the computation of M into multiple tableaus if any tableau (except the initial one) contains a run that continues from the previous tableau's first tail configuration.

We formalize the above as follows. Assume that T_α is a tableau, and that $T_\beta = \{w\}$ with $\ell(w) = (i,j)$ for some i. Then the formula $\psi_{\text{tail}}(\alpha, \beta)$ is meant to be true if and only if the i-th row of C_{T_α} is a tail configuration. Roughly speaking, with the parameters α and β we pass to $\psi_{\text{tail}}(\alpha, \beta)$ a tableau (viz. T_α) and the index of a row (viz. i). By using the shortcut

$$Q'\text{-state}(\beta) := \bigovee_{(q,a)\in Q'\times\Gamma} (\beta \hookrightarrow c^{-1}(q,a)),$$

we check if a given singleton $T_\beta = \{w\}$ encodes an accepting, rejecting, existential, universal, or an arbitrary state by setting Q' to Q_{acc}, Q_{rej}, $Q_\exists$, $Q_\forall$ or Q, respectively. As a result, we can define:

$$\psi_{\text{first-tail}}(\alpha, \beta) := \psi_{\text{tail}}(\alpha, \beta) \wedge {\sim}\exists^1_{\gamma_1} \left(\psi^{\mathfrak{t}}_{\prec}(\gamma_1, \beta) \wedge \psi_{\text{tail}}(\alpha, \gamma_1)\right)$$

$$\psi_{\text{tail}}(\alpha,\beta) := \exists^{\approx\alpha}_{\gamma_0}\, \exists^1_\alpha\, \psi^{\mathfrak{t}}_{\equiv}(\alpha,\beta) \wedge Q\text{-}\mathsf{state}(\alpha) \wedge \Big[Q_{\text{acc}}\text{-}\mathsf{state}(\alpha) \otimes Q_{\text{rej}}\text{-}\mathsf{state}(\alpha) \otimes$$
$$\exists^1_{\gamma_0}\Big(\psi^{\mathfrak{t}}_{\prec}(\gamma_0,\alpha) \wedge \big(Q_\exists\text{-}\mathsf{state}(\alpha) \wedge Q_\forall\text{-}\mathsf{state}(\gamma_0)\big) \otimes \big(Q_\forall\text{-}\mathsf{state}(\alpha) \wedge Q_\exists\text{-}\mathsf{state}(\gamma_0)\big)\Big)\Big]$$

▶ **Claim** (h). *Suppose that T_α is a tableau, $T_\beta = \{w\}$, and $\ell(w) = (i,j)$.*
Then $T \vDash \psi_{\textit{tail}}(\alpha,\beta)$ if and only if i is a tail index of C_{T_α}; and $T \vDash \psi_{\textit{first-tail}}(\alpha,\beta)$ if and only if i is the first tail index of C_{T_α}.

Formally, given a run C of M that has a tail configuration, C *accepts* if the state q in its first tail configuration is in Q_{acc}, C *rejects* if $q \in Q_{\text{rej}}$, and C *alternates* otherwise. That a run of the form C_{T_α} accepts resp. rejects is expressed by

$$\psi_{\text{acc}}(\alpha) := \exists^{\approx\alpha}_{\gamma_2}\, \exists^1_{\gamma_2}\, Q_{\text{acc}}\text{-}\mathsf{state}(\gamma_2) \wedge \psi_{\text{first-tail}}(\alpha,\gamma_2),$$
$$\psi_{\text{rej}}(\alpha) := \exists^{\approx\alpha}_{\gamma_2}\, \exists^1_{\gamma_2}\, Q_{\text{rej}}\text{-}\mathsf{state}(\gamma_2) \wedge \psi_{\text{first-tail}}(\alpha,\gamma_2).$$

In this formula, first the tableau T_α is copied to T_{γ_2} to extract with $\exists^1_{\gamma_2}$ the world carrying an accepting/rejecting state, while $\psi_{\text{first-tail}}(\alpha,\gamma_2)$ ensures that no alternation or rejecting/accepting state occurs at some earlier point in C_{T_α}. If the first tail configuration of the run contains an alternation, and if the run was existentially quantified, then it should be continued in a universally quantified tableau, and vice versa. The following formula expresses, given two tableaus T_α, T_β, that C_{T_β} is a *continuation* of C_{T_α}, i.e., that the first configuration of C_{T_β} equals the first tail configuration of C_{T_α}. In other words, if i is the first tail index of C_{T_α}, then $C_{T_\alpha}(i,j) = C_{T_\beta}(1,j)$ for all $j \in \{1,\ldots,N\}$.

$$\psi_{\text{cont}}(\alpha,\beta) := \exists^1_{\gamma_2}\, \psi_{\text{first-tail}}(\alpha,\gamma_2) \wedge \forall^1_\alpha \forall^1_\beta$$
$$\Big[\Big(\psi^{\mathfrak{t}}_{\min}(\beta) \wedge \psi^{\mathfrak{t}}_{\equiv}(\alpha,\gamma_2) \wedge \psi^{\mathfrak{p}}_{\equiv}(\alpha,\beta)\Big) \multimap \bigwedge_{e\in\Xi} (\alpha \vee \beta) \hookrightarrow {=}(e)\Big]$$

The above formula first obtains the first tail index i of C_{T_α} and stores it in a singleton $y \in T_{\gamma_2}$. Then for all worlds $w \in T_\alpha$ and $v \in T_\beta$, where v is $\mathfrak{t}$-minimal (i.e., in the first row) and w is in the same row as y, and which additionally agree on their $\mathfrak{p}$-component, the third line states that w and v agree on Ξ. Altogether, the i-th row of C_{T_α} and the first row of C_{T_β} then have to coincide.

The number of alternations is polynomially bounded, i.e., M performs at most $r(n)-1$ alternations for a polynomial r. In other words, we require at most $r = r(n)$ tableaus, which we call $\alpha_1,\ldots,\alpha_r$. In the following, the formula $\psi_{\text{run},i}$ describes the behaviour of the i-th run. W.l.o.g. r is even and $q_0 \in Q_\exists$. We may then define the final run by

$$\psi_{\text{run},r} := \forall^{\subseteq}_{\alpha_r}\Big[\Big(\psi_{\text{legal}}(\alpha_r) \wedge \psi_{\text{cont}}(\alpha_{r-1},\alpha_r)\Big) \multimap \Big(\sim\psi_{\text{rej}}(\alpha_r) \wedge \psi_{\text{acc}}(\alpha_r)\Big)\Big].$$

For $1 < i < r$ and even i, let

$$\psi_{\text{run},i} := \forall^{\subseteq}_{\alpha_i}\Big[\Big(\psi_{\text{legal}}(\alpha_i) \wedge \psi_{\text{cont}}(\alpha_{i-1},\alpha_i)\Big) \multimap \Big(\sim\psi_{\text{rej}}(\alpha_i) \wedge \big(\psi_{\text{acc}}(\alpha_i) \otimes \psi_{\text{run},i+1}\big)\Big)\Big]$$

and for $1 < i < r$ and odd i

$$\psi_{\text{run},i} := \exists^{\subseteq}_{\alpha_i}\Big[\psi_{\text{legal}}(\alpha_i) \wedge \psi_{\text{cont}}(\alpha_{i-1},\alpha_i) \wedge \sim\psi_{\text{rej}}(\alpha_i) \wedge \Big(\psi_{\text{acc}}(\alpha_i) \otimes \psi_{\text{run},i+1}\Big)\Big].$$

Analogously, the initial run is described by

$$\psi_{\text{run},1} := \exists^{\subseteq}_{\alpha_1}\Big(\psi_{\text{legal}}(\alpha_1) \wedge \psi_{\text{input}}(\alpha_1) \wedge \sim\psi_{\text{rej}}(\alpha_1) \wedge \Big(\psi_{\text{acc}}(\alpha_1) \otimes \psi_{\text{run},2}\Big)\Big)$$

Let us state the set $\Psi \subseteq \mathcal{PS}$ of all relevant scopes and the set $\Psi' \subseteq \Psi$ of scopes that accommodate pre-tableaus:

$$\Psi := \{\mathfrak{s}_i \mid 0 \leq i \leq k\} \cup \{\mathfrak{s}'_k\} \cup \{\gamma_i \mid 0 \leq i \leq n+1\} \cup \{\alpha_i \mid 1 \leq i \leq r\}$$
$$\Psi' := \{\gamma_i \mid 0 \leq i \leq n+1\} \cup \{\alpha_i \mid 1 \leq i \leq r\}$$

W.l.o.g. $n \geq 5$, as $\gamma_1, \ldots, \gamma_6$ are always used. Then we ultimately define

$$\varphi_x := \mathsf{canon}' \wedge \mathsf{scopes}_k(\Psi) \wedge \bigwedge_{p \in \Psi'} \psi_{\text{pre-tableau}}(p) \wedge \psi_{\text{run},1},$$

which is an MTL_k-formula since we deliberately omitted the conjunct $\Box^{k+1}\bot$ here. However, by Lemma 5.5, φ_x is satisfiable if and only if $\varphi_x \wedge \Box^{k+1}\bot$ is satisfiable. Finally, it is not hard using the above claims to prove that $\varphi_x \wedge \Box^{k+1}\bot$ is satisfiable if and only if M accepts x.