Technical Report: Hyperproperties in Matching Logic

Jan Tušil
Masaryk University, Brno
jan.tusil@mail.muni.cz
Xiaohong Chen and Grigore Roşu
University of Illinois at Urbana-Champaign
{xc3,grosu}@illinois.edu

Abstract

Matching logic is a uniform logic to specify and reason about programming languages and program properties. Many important logics and/or formal systems have been shown to be definable in matching logic as logical theories. However, no research has been conducted to studying how hyperproperties can be treated in matching logic. In this paper, we give the first theoretical result that shows that HyperLTL (hyper linear temporal logic), which is an important temporal logic designed for specifying and reasoning about hyperproperties, can be completely captured by matching logic. Our result demonstrates that matching logic offers a uniform treatment to handling hyperproperties and to supporting their model checking problems.

1 Introduction

A trace property, or simply a property, can be formally defined as a set of execution traces of a given system. We say that a system S satisfies a property P, if the set of all traces generated by S, denoted $tr(S)$, is included by P, that is, $tr(S) \subseteq P$. Trace properties have proven to be useful in specifying various dynamic properties of transition systems and there already exist many temporal logics such as linear temporal logic [25] (LTL) and computation tree logic [9] (CTL) that are specifically designed for expressing trace properties.

However, the rapid development and application of formal methods in the area of *security* shows that trace properties are not expressive enough to specify certain important dynamic properties. Typical examples include information flow policies, such as *noninterference* [11, 17] that states that security-sensitive data (such as passwords or private cryptographic keys) do not influence the outputs that is public to non-privileged users, privacy policies such as *data minimization principle*, [3, 24] that states that a system should collect only

data necessary for a given task, and performance requirements such as *average response time* [11].

Therefore, to support the specification and reasoning of the above security properties, researchers proposed *hyperproperties* [11]. A hyperproperty H is formally defined as a set of trace properties. In other words, H is a set of sets of execution traces. We say that a system S satisfies the hyperproperty H, if $tr(S) \in H$ (recall that the same satisfaction relation is defined as $tr(S) \subseteq P$ for a trace property P). Clearly, hyperproperties subsume trace properties: a trace property P can be regarded as the hyperproperty $H = \{P\}$ that is a singleton set.

Hyperproperties characterize the relationship among multiple execution traces as well as their interactions, and thus cannot be expressed using the existing temporal logics such as LTL or CTL. Therefore, researchers have proposed a number of new logics and/or formal systems that aim at supporting the specification and reasoning about hyperproperties. HyperLTL [10] is a typical and one of the first "hyper logics" that have been proposed, along with many others such as HyperCTL* [10] and Cartesian Hoare logic [29]. HyperLTL extends the classical LTL by adding (possibly alternative) trace quantifiers on top of an LTL formula. HyperLTL has been actively studied in literature. Its model checking algorithm was proposed in [16]. Its decidable fragments have been investigated in [22]. And there has been runtime verification technique proposed for HyperLTL [15]. All the above has made HyperLTL an important formalism for hyperproperties. In this paper, we target at HyperLTL and show that it can be completely defined in matching logic as a logical theory.

The *main motivation* of our research is to find a uniform treatment for hyperproperties using matching logic. Matching logic [5–7, 27] is a unifying logic to specify and reason about programs and their dynamic properties. Its syntax builds formulas, called *patterns*, which can express static structures, dynamic properties, and logical constraints in a uniform way; its semantics is based on *pattern matching*, where pattern *matches* a set of model elements. It has been shown that matching logic can define many important logical systems as its logical theories using patterns as axioms. Matching logic is also the logical foundational of the $\mathbb{K}$ formal language framework (`https://kframework.org`), in which the complete formal semantics of many real-world languages such as C, Java, and JavaScript have been formally defined [1, 19, 23].

The benefits of our work are closely related to the matching logic *vision*: that matching logic can serve as a logical foundation into which other logical systems can be "plugged" in a *modular* way. Under this vision, logical systems, computing systems, programming languages and programs are defined as theories which can be easily combined, and any logic L (such as HyperLTL) defined in matching logic can then be used to reason about any system defined in matching logic, be it a piece of software, hardware description, or a distributed system model. Formal reasoning within the logic L can be obtained using one fixed matching logic proof system, which we discuss in Section 2.2. With this work, this becomes true for HyperLTL, which, to the best of our knowledge, has no proof system.

Specifically speaking, we make the following technical contributions in the paper. Firstly, we define a matching logic theory written Γ^{hLTL} as well as an (almost verbatim) syntax translation function $\mathsf{H2M}$ from HyperLTL to matching logic. Secondly, we prove the following *semantic equivalence* theorem that states that Γ^{hLTL} indeed captures the semantics of HyperLTL:

Theorem 1 (Semantic Equivalence)**.** *Let φ be any HyperLTL formula and $\mathsf{H2M}(\varphi)$ its translation in matching logic. Then:*

$$\vDash_{\mathsf{hLTL}} \varphi \text{ iff } \Gamma^{\mathsf{hLTL}} \vDash \mathit{WellSorted}(\varphi) \rightarrow [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)$$

Intuitively, Theorem 1 states that φ is a valid HyperLTL formula if and only if the corresponding pattern $\mathsf{H2M}(\varphi)$ matches the set of all execution traces, denoted as $[\![\mathit{Trace}]\!]$ (see Section 5), given that all trace and state variables that appear in φ range over the intended sorts, as specified by the $\mathsf{WellSorted}(\varphi)$ predicate/pattern (see Section 5).

Theorem 1 connects the validity relations in HyperLTL and matching logic. However, in practical applications such as model checking, we are interested in not only HyperLTL validity but also its *models*. Does the proposed matching logic theory Γ^{hLTL} correctly capture HyperLTL models?

We give a positive answer to the above question in terms of the following two theorems, which we call *model equivalence* theorems. We show that there exist two model transformation functions, $\mathcal{M}^h$ (mapping HyperLTL models to matching logic) and $\mathcal{T}^h$ (mapping matching logic models to HyperLTL), such that the validity relations in both logics are invariant. Formally,

Theorem 2 (Model Equivalence A)**.** *For any HyperLTL model T and formula φ,*

$$T \vDash_{\mathsf{hLTL}} \varphi \text{ iff } \mathcal{M}^h(T) \vDash \mathit{WellSorted}(\varphi) \rightarrow [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)$$

Theorem 3 (Model Equivalence B)**.** *For any matching logic model $M \vDash \Gamma^{\mathsf{hLTL}}$ and HyperLTL formula φ,*

$$\mathcal{T}^h(M) \vDash_{\mathsf{hLTL}} \varphi \text{ iff } M \vDash \mathit{WellSorted}(\varphi) \rightarrow [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)$$

The above model equivalence theorems allow us to use matching logic, including its full syntax of patterns (and not only those corresponding to HyperLTL formulas) to specify HyperLTL models, and to use the matching logic proof system (see Section 2) to reason about them. One such application is to support the problem of *model checking* for HyperLTL, which we discuss in detail in Section 6.

It is worthwhile to mention that our construction of the matching logic theory Γ^{hLTL} is directly based on the previous work that defines LTL in matching logic in [6]. Specifically, we first extend the work on LTL from [6] into theory Γ^{LTL} which satisfies model equivalence theorems, and then extend Γ^{LTL} into Γ^{hLTL}.

In the following, we first introduce matching logic, including its syntax, semantics, proof system, and logical theories in Sections 2 and 3. Then,

- We review the definition of LTL in matching logic and prove two new results—the model equivalence theorems for LTL (Theorem 9 and Theorem 15) in Section 4.
- We extend the definition of LTL to that of HyperLTL in an easy and intuitive way. Then, we prove the model equivalence theorems for HyperLTL (Theorems 2 and 3) in Section 5, from which the semantic equivalence theorem (Theorem 1) is proved in 6.
- We discuss the practical application of the proposed theory Γ^{hLTL} in HyperLTL validity and/or model checking in Section 6.

Finally, we conclude the paper with related work in Sections 7 and 8.

2 Preliminaries About Matching Logic

In this section we formally introduce the syntax and semantics of matching logic. We refer interested readers to [5–7, 27] for details.

2.1 Matching Logic Syntax and Semantics

Matching logic is a seamless combination of first-order logic (FOL) and modal μ-logic [20] that makes no distinction among functions, predicates, or modal operators, but uses *symbols* to uniformly construct *patterns*, which capture static structures, dynamic properties, and logical constraints. It is parametric in two variable sets: (1) the set EV of *element variables*, which are FOL-style variables that evaluate to individual elements; and (2) the set SV of *set variables* that evaluate to sets. Given a matching logic *signature* Σ as a set of *symbols*, the syntax of matching logic *patterns* [5] is inductively defined:

$$\begin{aligned}\varphi ::= {} & x \mid X \mid \sigma \mid \varphi_1\, \varphi_2 \mid \bot \mid \varphi_1 \to \varphi_2 \\ & \mid \exists x \,.\, \varphi \mid \mu X \,.\, \varphi \quad \text{where } \varphi \text{ is positive in } X\end{aligned}$$

Here, x ranges over element variables in EV, X over set variables in SV, and σ over symbols in Σ. The pattern φ is positive in X, if no free occurrences of X are nested an odd number of times on the left of an implication $\varphi_1 \to \varphi_2$. The construct $\varphi_1\, \varphi_2$ is called *application* and is left-associative. Often, application has the form $\sigma\, \varphi$, where $\sigma \in \Sigma$ is a symbol. We use $\textsc{Pattern}(\Sigma)$ to denote the set of all patterns generated by the above grammar, and write $\varphi[\psi/x]$ to mean the result of substituting ψ for x in φ while avoiding free variable capture. For notational simplicity, we define the syntactic sugar as usual:

$$\begin{aligned} \neg\varphi &\equiv \varphi \to \bot & \qquad \varphi_1 \vee \varphi_2 &\equiv \neg\varphi_1 \to \varphi_2 \\ \top &\equiv \neg\bot & \qquad \varphi_1 \wedge \varphi_2 &\equiv \neg(\neg\varphi_1 \vee \neg\varphi_2) \\ \forall x \,.\, \varphi &\equiv \neg\exists x \,.\, \neg\varphi & \qquad \nu X \,.\, \varphi &\equiv \neg\mu X \,.\, \neg\varphi[\neg X/X] \end{aligned}$$

The *semantics* of matching logic is based on *pattern matching*: patterns are interpreted as the sets of elements that *match* them. For example, $\varphi_1 \wedge \varphi_2$ is matched by elements that match both φ_1 and φ_2; $\varphi_1 \vee \varphi_2$ is matched by elements that match φ_1 or φ_2, etc. Element variable $x \in \mathrm{EV}$ is matched by one single element, yielding the same semantics as FOL variables; $\exists x \,.\, \varphi$ builds *abstraction*, which is matched by all elements that can match φ for some valuations of x; $\mu X \,.\, \varphi$ builds a *least fixpoint pattern*, which is matched by the elements in the smallest set X such that $X = \varphi$ (note that X may occur recursively in φ).

Formally, a matching logic *model* consists of

1. a nonempty carrier set M;
2. a binary function $app_M \colon M \times M \to \mathcal{P}(M)$ (where $\mathcal{P}(M)$ denotes the *powerset* of M) as the *interpretation of application*;
3. and a subset $\sigma_M \subseteq M$ as the interpretation of $\sigma \in \Sigma$.

For notational simplicity, we use M to denote both the carrier set and the whole model. We extend application from elements to sets: $appext_M(A, B) = \bigcup_{a \in A, b \in B} app_M(a, b)$ for $A, B \subseteq M$. The semantics of patterns is given with respect to a *variable valuation* $\rho \colon (\mathrm{EV} \cup \mathrm{SV}) \to (M \cup \mathcal{P}(M))$ that maps element variables to model elements and set variables to sets of model elements. The *interpretation* of a pattern φ, denoted as $|\varphi|_{M,\rho}$ or simply $|\varphi|_\rho$, is a set of model elements, defined inductively as:

- $|x|_{M,\rho} = \{\rho(x)\}$ for $x \in \mathrm{EV}$
- $|X|_{M,\rho} = \rho(X)$ for $X \in \mathrm{SV}$
- $|\bot|_{M,\rho} = \emptyset$
- $|\varphi_1 \to \varphi_2|_{M,\rho} = M \setminus (|\varphi_1|_{M,\rho} \setminus |\varphi_2|_{M,\rho})$
- $|\sigma|_{M,\rho} = \sigma_M$ for $\sigma \in \Sigma$
- $|\varphi_1\, \varphi_2|_{M,\rho} = appext_M(|\varphi_1|_{M,\rho}, |\varphi_2|_{M,\rho})$
- $|\exists x \,.\, \varphi|_{M,\rho} = \bigcup_{a \in M} |\varphi|_{M,\rho[a/x]}$
- $|\mu X \,.\, \varphi|_{M,\rho} = \mathbf{lfp}(A \mapsto |\varphi|_{M,\rho[A/X]})$

where "$\setminus$" denotes set difference; $\rho[a/x]$ (resp. $\rho[A/X]$) denotes valuation update, which is the valuation ρ' such that $\rho'(x) = a$ (resp. $\rho'(X) = A$) and agrees with ρ on all other variables; **lfp** denotes the true least fixpoint, since $A \mapsto |\varphi|_{M,\rho[A/X]}$ is provably monotone (Lemma 26 in the appendix) and thus have a unique least fixpoint by the Knaster-Tarski fixpoint theorem [30].

Definition 4. *A model* M satisfies *a formula* φ*, written* $M \vDash \varphi$*, iff the formula is interpreted as the whole set* M *in all valuations; that is, iff* $|\varphi|_\rho = M$ *for all* ρ*. A theory* Γ*, which is a set of matching logic patterns called* axioms*, is satisfied in a model* M*, written* $M \vDash \Gamma$*, iff* $M \vDash \varphi$ *for all* $\varphi \in \Gamma$*. We define*

$\Gamma \vDash \varphi$ *iff* $M \vDash \varphi$ *for all* $M \vDash \Gamma$*, and let* $\mathbf{Mod}_{\mathsf{ML}}(\Gamma) = \{M \mid M \vDash \Gamma\}$ *be the class of all models of the theory* Γ*. For any ML pattern* φ*, let* $FV(\varphi)$ *be the set of all free variables of* φ*. When* $FV(\varphi) = \emptyset$*, the interpretation of* φ *does not depend on the valuation, so we sometimes use the notation* $|\varphi|_M$ *to denote the unique interpretation of* φ *in the model* M*.*

2.2 Matching Logic Proof System

Matching logic has a Hilbert-style proof system that defines the provability relation $\Gamma \vdash \phi$, which means that φ can be proved in the proof system, where the patterns from Γ are added as additional axioms [6,7]. The following theorem states that the proof system is sound.

Theorem 5 ([6], Theorem 24)**.** $\Gamma \vdash \varphi$ *implies* $\Gamma \vDash \varphi$

Since the proof system is sound for reasoning in any matching logic theory, it is also sound for reasoning in all theories presented in this paper, including theories Γ^{LTL} and Γ^{hLTL} that we present in Sections 4 and 5 - see discussion in Section 6.

3 Example Matching Logic Theories

Matching logic is simple, and does not have many important instruments built into it. In particular, it does not have equality; it has symbols, but not function symbols; it does not have sorts; it has μ, but does not have multi-ary recursive symbols. All these instruments would be useful for defining LTL and HyperLTL in matching logic. Fortunately, all these features can be defined using axioms and notations, without extending the logic. In this section we show how it can be done; we will use the instruments defined here in the sections that follows, when we define LTL in Section 4 HyperLTL in Section 5.

3.1 Equality, Inclusion, Membership

We show how to define equality $\varphi_1 = \varphi_2$, set inclusion $\varphi_1 \subseteq \varphi_2$, and membership $x \in \varphi$ in an axiomatically way in matching logic as a logical theory. Specifically, we define the equality $\varphi_1 = \varphi_2$ (similar for inclusion and membership) as a matching logic pattern such that it holds (i.e., evaluates to $\top$), iff φ_1 evaluates to the same set as φ_2, and it fails (e.g., evaluates to $\bot$) otherwise.

We define equality, inclusion, and membership as shown in Spec. 1. The specification defines a matching logic signature $\Sigma^{\mathsf{DEFINEDNESS}}$ containing one symbol, called "definedness", a theory $\Gamma^{\mathsf{DEFINEDNESS}}$ containing one axiom, called (Definedness), and a few notations, which allow us to write, e.g., $\lceil\varphi\rceil$ instead of $\lceil\_\rceil\,\varphi$. The axiom (Definedness) enforces that in all models, $|\lceil\varphi\rceil|_\rho = M$ iff $|\varphi|_\rho \neq \emptyset$. (It also holds that $|\lceil\varphi\rceil|_\rho = \emptyset$ otherwise, even without the axiom.) We use the name "definedness" for the symbol and the axiom, since the pattern $\lceil\varphi\rceil$ means that φ is defined; i.e., matched by at least by one element. With

spec DEFINEDNESS
Symbol: $\lceil \_ \rceil$
Notation:

$$\lceil \varphi \rceil \equiv \lceil \_ \rceil\, \varphi \qquad \lfloor \varphi \rfloor \equiv \neg \lceil \neg \varphi \rceil$$
$$\varphi_1 = \varphi_2 \equiv \lfloor \varphi_1 \leftrightarrow \varphi_2 \rfloor \qquad \varphi_1 \neq \varphi_2 \equiv \neg(\varphi_1 = \varphi_2)$$
$$x \in \varphi \equiv \lceil x \wedge \varphi \rceil \qquad \varphi_1 \subseteq \varphi_2 \equiv \lfloor \varphi_1 \leftrightarrow \varphi_2 \rfloor$$
$$x \notin \varphi \equiv \neg(x \in \varphi) \qquad \varphi_1 \not\subseteq \varphi_2 \equiv \neg(\varphi_1 \subseteq \varphi_2)$$

Axiom:
(DEFINEDNESS) $\lceil x \rceil$
endspec

Spec. 1: Definedness and related notions

the definedness symbol and axiom, we can define equality, membership, and set inclusion, as syntactic sugar, using another notation, $\lfloor \varphi \rfloor$, called "totality". Intuitively, $\lfloor \varphi \rfloor$ states that φ is matched by all elements - we say that φ is "total". Then, $\varphi_1 = \varphi_2$ states that φ_1 and φ_2 are matched by the same elements, etc. Note that the above is not an extension of matching logic, but merely one symbol in the signature, one axiom, and a few notations. Equality, membership and inclusion defined this way have the intended semantics:

Proposition 6. *Let M be a $\Gamma^{\mathsf{DEFINEDNESS}}$-model. Then for any M-valuation ρ and any patterns φ_1, φ_2,*

1. $|\varphi_1 = \varphi_2|_\rho$ *iff* $|\varphi_1|_\rho = |\varphi_2|_\rho$;
2. $|\varphi_1 \subseteq \varphi_2|_\rho$ *iff* $|\varphi_1|_\rho \subseteq |\varphi_2|_\rho$;
3. $|\varphi_1 \in \varphi_2|_\rho$ *iff* $m \in |\varphi_2|_\rho$ *whenever* $\{m\} = |\varphi_1|_\rho$ *for some* $m \in M$.

Unlike FOL, where formulas evaluate to either true or false, matching logic patterns can evaluate to any sets. However, some patterns, such as $\varphi_1 = \varphi_2$, can evaluate only to the empty set or the total set. We call such patterns *predicate patterns*. Intuitively, the purpose of predicate patterns is to make a statement. If a predicate pattern evaluates to the empty set, it means that the statement is false, while if it evaluates to the total set, the statement is true. So for example, the pattern $\lceil \varphi \rceil$ make the statement that φ evaluates to a nonempty set (i.e., matches something); the pattern $x \in \varphi$ says that φ matches x. Predicate patterns are closed over boolean connectives and quantification, and they correspond the first-order logic formulas. We often use predicate patterns to specify a condition on elements being matched. For example, the pattern $x \wedge (x \neq y)$ can match any element distinct from y.

spec SORTS
  Import: DEFINEDNESS
  Symbol: $inh, Sort$
  Notation:
    $[\![s]\!] \equiv inh\ s$
    $\neg_s \varphi \equiv (\neg\varphi) \wedge [\![s]\!]$
    $\forall x{:}s\,.\,\varphi \equiv \forall x\,.\,x \in [\![s]\!] \to \varphi$
    $\exists x{:}s\,.\,\varphi \equiv \exists x\,.\,x \in [\![s]\!] \wedge \varphi$
endspec

Spec. 2: Sorts and sorted quantification

3.2 Sorts

In Sections 4 and 5 we need to represent (Hyper)LTL traces, quantify over them and define functions on them. For this purpose, we use *sorts*. Although matching logic has no builtin support for sorts or many-sorted functions, it can represent them by symbols defined with proper axioms. Specifically, for every sort s, we define a corresponding symbol also denoted s to represent its name, and define a symbol $[\![\_]\!]$, called *inhabitant*, with the intuition that $[\![s]\!]$ is matched by all elements of sort s. Then, we can specify properties about sorts by patterns; e.g.: (Nonempty Inhabitant) $[\![s]\!] \neq \bot$ specifies that the inhabitant of the sort s is nonempty. We define *sorted quantification* that requires x to have sort s in Spec. 2.

3.3 Many-sorted Functions

As we have seen, the pattern $\sigma\, x_1 \ldots x_n$ can be matched by zero, one, or more elements. In practice, we often need symbols to be interpreted as functions (or partial functions), which are special instances where $\sigma\, x_1 \ldots x_n$ is matched by exactly (or at most) one element. Axiomatically, we define a many-sorted function $f\colon s_1 \times \cdots \times s_n \to s$ as:

$$\text{(Function)}\quad \forall x_1{:}s_1 \ldots \forall x_n{:}s_n\,.\,\exists y{:}s\,.\,f\, x_1 \ldots x_n = y$$

which enforces $f\, x_1 \ldots x_n$ to return exactly one element y of sort s, given that $x_1, \ldots, x_n$ have the appropriate sorts. We use the notation $\epsilon \to s$ to denote a nullary function of sort s, i.e., a constant. Partial functions $f\colon s_1 \times \cdots \times s_n \rightharpoonup s$ are axiomatized similarly: replace $f\, x_1 \ldots x_n = y$ in (Function) with $f\, x_1 \ldots x_n \subseteq y$, which enforces $f\, x_1 \ldots x_n$ to be at most one element.

3.4 Natural Numbers

With the μ binder we can axiomatize natural numbers as the smallest set that contains *zero* and is closed under *succ* - see the axiom (NatDomain) in Spec. 3.

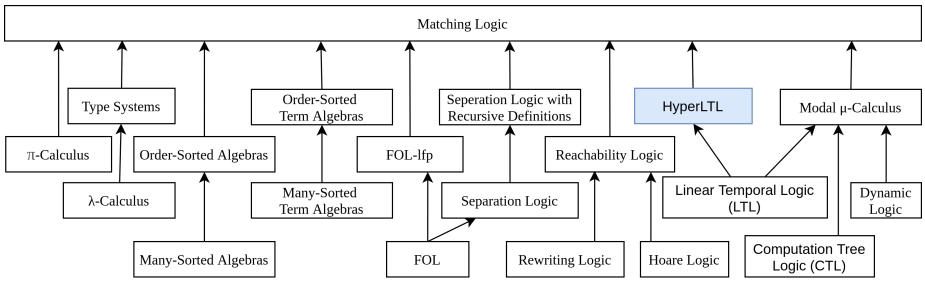


Figure 1: Many important logics can be defined as theories in matching logic.

Recall that the operator μ is interpreted as the least fixpoint of a function

spec NAT
Import: SORTS
Symbol: $Nat, zero, succ$
Axiom:

(ZEROF)	$zero : \epsilon \to Nat$
(SUCCF)	$succ : Nat \to Nat$
(NATDOMAIN)	$[\![Nat]\!] = \mu X.\, zero \vee succ\, X$
(NATNOCONF)	$\forall x : Nat,\ zero \neq succ\, x$
(SUCCINJ)	$\forall x, y : Nat, succ\, x = succ\, y \to x = y$

endspec

Spec. 3: Natural numbers

determined by the body of the μ pattern and the variable bound by the μ operator. In this case, $[\![Nat]\!]$ is defined as the least fixpoint of the function

$$A \mapsto zero_M \cup appext_M(succ_M, A)\,,$$

because the symbol $\vee$ in the pattern $zero \vee succ\, X$ gets interpreted as the set union, the application in $succ\, X$ as the pointwise-extended application defined in the model, the symbols $zero$ and $succ$ as $zero_M$ and $succ_M$, respectively, and the variable X as A.

3.5 Pairs

Spec. 4 defines pairs with the pair constructor $\langle \_, \_ \rangle$ and projections $proj_1$ and $proj_2$. Furthermore, we define an axiom (KEYVALUE) for pair application, so that a pair applied to an element returns the second component of a pair whenever the first component matches the argument.

spec PAIR
Symbol: $pair, proj_1, proj_2$
Notation:
$\langle x, y\rangle \equiv pair\ x\ y$
Axiom:

(PairFunction)	$\forall x.\, \forall y.\, \exists z.\, \langle x, y\rangle = z$
(PairInjective)	$\langle x_1, y_1\rangle = \langle x_2, y_2\rangle$ $\rightarrow x_1 = x_2 \wedge y_1 = y_2$
(ProjectPair1)	$\forall x_1.\, \forall x_2.\, proj_1\ \langle x_1, x_2\rangle = x_1$
(ProjectPair2)	$\forall x_1.\, \forall x_2.\, proj_2\ \langle x_1, x_2\rangle = x_2$
(KeyValue)	$\langle k_1, v\rangle\ k_2 = ((k_1 = k_2) \wedge v)$

endspec

Spec. 4: Pairs and their axioms

3.6 Recursive Symbols

Recursive symbols are symbols that are interpreted as *the least solution* of recursive equations $f\, x_1 \ldots x_n =_{\text{lfp}} \varphi$ where f occurs positively in φ (most likely with different arguments). Such recursive symbols can be defined using the existing μ-binder in matching logic, as shown in Spec. 5 for binary recursive symbols. The intuition is that the μ term builds a graph of the function, i.e., the set of argument-result pairs, from which the appropriate results are selected using application and the axiom (KeyValue). In the binary case, the result is then another argument-result graph, which is then searched again for the second argument. The notation $f\, x_1 \ldots x_n =_{\text{gfp}} \varphi$ is used similarly to define symbols that are interpreted as *the greatest solution* of a recursive equation. We will see concrete examples of recursive symbols in Section 5.2.

spec RECURSION
Import: PAIR
Notation:
$f\ x_1\ x_2 =_{\text{lfp}} \varphi$
$\equiv f\ x_1\ x_2 = (\mu\ f.\, \exists x_1.\, \exists x_2.\, \langle x_1, \langle x_2, \varphi\rangle\rangle)\ x_1\ x_2$
$f\ x_1\ x_2 =_{\text{gfp}} \varphi$
$\equiv f\ x_1\ x_2 = (\nu\ f.\, \exists x_1.\, \exists x_2.\, \langle x_1, \langle x_2, \varphi\rangle\rangle)\ x_1\ x_2$
endspec

Spec. 5: Recursive symbols as notations over μ and pairs

3.7 Logical Systems as Theories

We have seen above that matching logic is powerful enough to define equality, sorts, functions, natural numbers, pairs and recursion, all using only axioms and notations, without extending the logic itself. In the same style, matching logic can capture whole logical systems. In Fig. 1 we show a selection of logical systems that have been formalized as theories in matching logic.

We are not arguing that matching logic is in some sense better than any of the other systems from Fig. 1. These results mean only that matching logic can serve as a unifying foundation into which other logics can be integrated. This is useful, since it allows specification of and reasoning about systems using a variety of logics, each specialized to its own domain. In the following sections, when we define LTL and HyperLTL as theories in matching logic, our aim is not to replace (Hyper)LTL with matching logic. Instead, we want to bring HyperLTL reasoning into one unifying framework that allows reasoning using various logics about programs written in various languages.

4 Defining LTL in Matching Logic

In this section we review linear temporal logic (LTL) [26], define it as a matching logic specification LTL, define translations from LTL to matching logic models and back, and show that our definitions are well-behaved by proving model equivalence theorems. We also compare our axiomatization of LTL with those of [6], which is simpler and does not capture models. However, the main purpose of this section is to demonstrate the approach that we use for HyperLTL in Section 5, on a simpler logic. In this section, as well as in Section 5, we focus on conveying the intuition behind our constructions; we refer an interested reader to the Appendix.

4.1 LTL Syntax and Semantics

The *syntax* of LTL is parametric in a countable set AP of *atomic propositions* denoted as a; from now on, let AP be fixed. The set $\mathbf{\Phi}_{\mathsf{LTL}}$ of LTL *formulas* is defined by the following grammar:

$$\psi ::= a \mid \neg\psi \mid \psi \wedge \psi \mid \circ\psi \mid \psi \, U \, \psi$$

Intuitively, $\circ\psi$ holds in a state of a path iff ψ holds in the next state of the path (path is an infinite sequence of states); $\psi_1 \, U \, \psi_2$ holds on a state of a path iff ψ_2 holds in some future state of that path and all states until that point satisfy ψ_1. An *infinite trace* $\tau \in (\mathcal{P}(\mathrm{AP}))^\omega$, or simply *trace*, is an infinite sequence of subsets of atomic propositions. We write $\tau[i]$ to mean the ith (starting from 1) element of trace τ for $i \geq 1$. We use $\tau[i..]$ to denote the suffix trace $\tau[i], \tau[i+1], \ldots$.

LTL models are infinite traces. We let $\mathbf{Mod}_{\mathsf{LTL}} = (\mathcal{P}(\mathrm{AP}))^\omega$ denote the set of all LTL models. The semantics of LTL formula is defined w.r.t. a trace τ as the relation $\models_{\mathsf{LTL}} \subseteq \mathbf{Mod}_{\mathsf{LTL}} \times \mathbb{N}_{\geq 1} \times \mathbf{\Phi}_{\mathsf{LTL}}$ inductively defined as follows:

- $\tau, i \vDash_{\mathsf{LTL}} a$ iff $a \in \tau[i]$;
- $\tau, i \vDash_{\mathsf{LTL}} \neg\psi$ iff $\tau, i \nvDash_{\mathsf{LTL}} \psi$;
- $\tau, i \vDash_{\mathsf{LTL}} \psi_1 \wedge \psi_2$ iff $\tau, i \vDash_{\mathsf{LTL}} \psi_1$ and $\tau, i \vDash_{\mathsf{LTL}} \psi_2$;
- $\tau, i \vDash_{\mathsf{LTL}} \circ\psi$ iff $\tau, i+1 \vDash_{\mathsf{LTL}} \psi$;
- $\tau, i \vDash_{\mathsf{LTL}} \psi_1 \, U \, \psi_2$ iff there exists $j \geq i$ such that $\tau, j \vDash_{\mathsf{LTL}} \psi_2$ and for all $i \leq k < j$ we have $\tau, k \vDash_{\mathsf{LTL}} \psi_1$.

We write $\tau \vDash_{\mathsf{LTL}} \psi$ if $\tau, 1 \vDash_{\mathsf{LTL}} \psi$ and $\vDash_{\mathsf{LTL}} \psi$ if $\tau \vDash_{\mathsf{LTL}} \psi$ for all $\tau \in (\mathcal{P}(\mathrm{AP}))^\omega$.

4.2 Capturing LTL in Matching Logic

We define a matching logic specification LTL that captures precisely LTL models, shown in Spec. 6; the specification defines a signature Σ^{LTL} and a Σ^{LTL}-theory Γ^{LTL}. We explain the specification as follows. Models of LTL formulas are traces; therefore, we introduce a sort *Trace* whose only inhabitant is a trace from $(\mathcal{P}(\mathrm{AP}))^\omega$. To determine whether a trace satisfies a LTL formula, one needs to consider the suffixes of the trace. A trace suffix is, intuitively, a trace paired with an offset; We represent trace suffixes using the sort *TrSuf*, and identify (full) traces with trace suffixes paired with 1. The intuition is that a Σ^{LTL}-pattern representing an LTL formula matches exactly those trace suffixes that satisfy the formula. We also include a symbol a for every atomic proposition; the symbol is intended to match any trace suffix τ, i whose first state $\tau[i]$ satisfies the atomic proposition. The behavior of this symbol is axiomatized by the axiom schema (AtomicProp), where the metavariable a ranges over all atomic propositions.

Now we explain the main operators/notations that we define; namely, $\neg_{\mathsf{LTL}}$, $\circ$, $\bar{\circ}$, and U.

1. The notation $\neg_{\mathsf{LTL}}$ represents negation of an LTL formula. We cannot use the built-in matching logic negation for the purpose of negating an LTL formula, since the carrier set may contain elements other than trace suffixes, and these would be included in the result of negation. For example, in some model, the pattern $\neg a$ matches the definedness symbol.
2. The symbol $\circ$ represents the LTL "next" operator. The intended meaning of this operator is the following: the pattern $\circ\varphi$ matches ("holds in") the trace suffix τ, i iff φ matches ("holds in") $\tau, i+1$. See the following illustration:

τ, i	$\tau, i+1$	$\tau, i+2$	*// trace suffixes*
$\circ\circ\varphi$	$\circ\varphi$	φ	*// patterns*

Therefore, perhaps counterintuitively, we can view the operator $\circ$ ("next") as a function in the *opposite* direction: if it takes as an argument $\tau, i+1$, it

spec LTL
Import: SORTS
Symbol: Trace, TrSuf, $\circ$, $\bar{\circ}$,
atomic proposition a *(for every* $a \in \mathrm{AP}$*)*
Notation:
$\neg_{\mathsf{LTL}}\varphi \equiv [\![\mathit{TrSuf}]\!] \wedge \neg\varphi$
$\varphi_1 \, U \, \varphi_2 \equiv \mu X \,.\, \varphi_2 \vee (\varphi_1 \wedge \circ X)$
Axiom:

(Prev)	$\bar{\circ}x = \exists y \,.\, y \wedge (x \in \circ y))$
(Trace)	$\exists x. [\![\mathit{Trace}]\!] = x$
(TraceSuffix)	$[\![\mathit{TrSuf}]\!] = \mu X. [\![\mathit{Trace}]\!] \vee \bar{\circ} X$
(Inf)	$[\![\mathit{TrSuf}]\!] \subseteq \circ [\![\mathit{TrSuf}]\!]$
(NextOut)	$\circ(\neg [\![\mathit{TrSuf}]\!]) \subseteq \neg [\![\mathit{TrSuf}]\!]$
(NextPFun)	$\circ \colon \mathit{TrSuf} \rightharpoonup \mathit{TrSuf}$
(NextInj)	$\forall x_1, x_2 \colon \mathit{TrSuf} \,.\, \circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot$ $\rightarrow x_1 = x_2$
(AtomicProp)	$a \subseteq [\![\mathit{TrSuf}]\!]$

endspec

Spec. 6: LTL as a matching logic specification

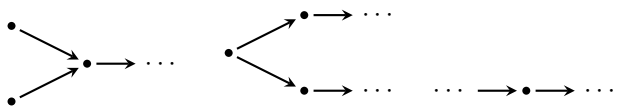
returns τ, i - intuitively, it extends the trace suffix given as the argument with the "previous" state $\tau[i]$. Consequently, one can obtain the (full) trace $\tau, 1$ of a trace suffix τ, i by repeatedly applying $\circ$.

3. The operator $\bar{\circ}$ ("previous") is a symbol defined by the axiom (Prev) to be the inverse of $\circ$; i.e., given a trace suffix τ, i as the argument, $\bar{\circ}$ returns $\tau, i+1$. The axiom intuitively says that $\bar{\circ}z$ returns the set of all traces for which $\circ$ yields z. In matching logic, there exists an idiomatic way to express the set of all elements that satisfy a given property (i.e., set comprehension): the scheme $\exists x. \, x \wedge P(x)$, where $P(x)$ is a predicate. Recall that in matching logic, existential quantifier is interpreted as a union; only elements satisfying given predicate contribute to the union, because the other elements get filtered out by intersection with the failing predicate, i.e., the empty set.

4. The operator U ("until") is defined as a notation over $\circ$. Recall that $\varphi_1 \, U \, \varphi_2$ intuitively means that "eventually, φ_2 holds, and until that point, φ_1 holds". This operator has a recursive nature, since we can expand:

$$\varphi_1 \, U \, \varphi_2 \simeq \varphi_2 \vee (\varphi_1 \wedge (\varphi_1 \, U \, \varphi_2)) \,.$$

By using $\simeq$ we want to say that the two formulas are semantically equivalent. Intuitively, the pattern $\mu X. \, \varphi_2 \vee (\varphi_2 \wedge \circ X)$ implements U using this expansion.



(a) merging (b) branching (c) infinite past

Figure 2: Models prohibited by Γ^{LTL}.

A question might arise, why does a model $\Gamma^{\mathsf{LTL}} \vDash M$ contain any trace suffixes at all? Intuitively, the axiom (Trace) ensures that the model contains a trace; because of the axiom (TraceSuffix), this trace is also a trace suffix; and the axiom (Inf) ensures we can generate trace suffixes *ad infinitum*. But then, why do we need the other axioms, and why (TraceSuffix) does not simply state that $[\![\mathit{Trace}]\!] \subseteq [\![\mathit{TrSuf}]\!]$? This is because in order for model elements to faithfully represent trace suffixes, we need to have some structure on the elements. (TraceSuffix) says that the sort of all trace suffixes is generated from the sort of traces by applying the operator $\bar{\circ}$; that is, by repeatedly adding 1 to the offset. This way, we ensure *finite past*. The axiom (NextOut) ensures that only trace suffixes can be predecessors of trace suffixes; the axioms (NextPFun) and (NextInj) ensure that $\circ$ is interpreted as an injective partial function, thus making sure that trace suffixes do not "branch" or "merge". For an illustration of models prohibited by Γ^{LTL}, see Fig. 2, where an arrow between x and y means that $x \in \mathit{appext}_M(\circ_M, \{y\})$.

From LTL Formulas to Matching Logic Patterns

With the symbols and notations in place, we can now define a formula translation function $\mathsf{L2M} \colon \mathbf{\Phi}_{\mathsf{LTL}} \to \textsc{Pattern}(\Sigma^{\mathsf{LTL}})$ from LTL formulas to matching logic formulas. The translation function is straightforward, directly mapping LTL constructs to symbols and notations of the specification LTL:

- $\mathsf{L2M}(a) = a$ for all atomic propositions a;
- $\mathsf{L2M}(\neg\psi) = \neg_{\mathsf{LTL}}\mathsf{L2M}(\psi)$;
- $\mathsf{L2M}(\psi_1 \wedge \psi_2) = \mathsf{L2M}(\psi_1) \wedge \mathsf{L2M}(\psi_2)$;
- $\mathsf{L2M}(\circ\psi) = \circ\,\mathsf{L2M}(\psi)$; and
- $\mathsf{L2M}(\psi_1\, U\, \psi_2) = \mathsf{L2M}(\psi_1)\, U\, \mathsf{L2M}(\psi_2)$.

For any Γ^{LTL}-model M, by axiom (Trace), $|[\![\mathit{Trace}]\!]|_M$ is a singleton set; we denote its unique element as M_{Trace}. We also write M_{TrSuf} to mean $|[\![\mathit{TrSuf}]\!]|_M$. Intuitively, M_{Trace} is the trace represented by the model M, while M_{TrSuf} represents all the suffixes of the trace M_{Trace}. The translation function $\mathsf{L2M}$ has the following property:

Lemma 7. *For a Γ^{LTL}-model M and an LTL formula φ, its corresponding matching logic translation $\mathsf{L2M}(\varphi)$ is matched only by trace suffixes; i.e.,*

$$|\mathsf{L2M}(\varphi)|_M \subseteq M_{TrSuf} \,.$$

4.3 From LTL Models to Matching Logic Γ^{LTL}-Models

In this section we prove the first model equivalence theorem, which says that an LTL model satisfies a formula if and only if its translation to matching logic satisfies the translated formula. We defer the concrete construction of the model translation function

$$\mathcal{M}^l : \mathbf{Mod}_{\mathsf{LTL}} \to \mathbf{Mod}_{\mathsf{ML}}(\Gamma^{\mathsf{LTL}})$$

to appendix (see Definition 30). Here, we only need to know that the carrier set of the constructed model contains positive natural numbers: $\mathcal{M}^l(\tau)_{TrSuf} = \mathbb{N}_{\geq 1}$ for any LTL model (i.e., a trace) τ, and that $\circ$ is interpreted as a partial function decrementing the number it is given and $\bar{\circ}$ as a total function incrementing the number it is given.

We said earlier that the intuition behind the sort *TrSuf* is that a Σ^{LTL}-pattern representing an LTL formula matches exactly those trace suffixes that satisfy the formula. The following lemma makes the intuition precise, thus connecting the semantics of LTL and matching logic.

Lemma 8. *$\tau, i \vDash_{\mathsf{LTL}} \varphi \iff i \in |\mathsf{L2M}(\varphi)|_{\mathcal{M}^l(\tau)}$ for any $\tau \in \mathbf{Mod}_{\mathsf{LTL}}$, $i \geq 1$, and a LTL formula φ.*

Now we would like to prove Theorem 9, which connects LTL validity with matching logic validity. However, we must be careful when formulating such proposition, because the pattern $\mathsf{L2M}(\varphi)$ is matched only by trace suffixes (Lemma 7), while a matching logic pattern is valid if it matches all elements. Since we only want the pattern $\mathsf{L2M}(\varphi)$ to match the (full) trace, we state:

Theorem 9 (Model Equivalence A)**.** *For any LTL model $\tau \in \mathbf{Mod}_{\mathsf{LTL}}$ and an LTL formula φ,*

$$\tau \vDash_{\mathsf{LTL}} \varphi \iff \mathcal{M}^l(\tau) \vDash [\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi) \,.$$

Intuitively, this theorem says that τ satisfies φ in LTL iff the translated formula matches the $[\![\mathit{Trace}]\!]$ component of the model $\mathcal{M}^l(\tau)$; i.e., the element M_{Trace}. A reader might wonder why the theorem does not say that $[\![\mathit{Trace}]\!] = \mathsf{L2M}(\varphi)$, which would intuitively mean that φ matches exactly the trace. The answer is that such theorem would not hold in general, since the translated formula may match many trace suffixes, i.e., inhabitants of the sort *TrSuf*, of which the full trace is only one special member.

4.4 From Matching Logic Γ^{LTL}-Models to LTL Models

In this section we prove Theorem 15, which says, intuitively, that any matching logic model of Γ^{LTL} can be transformed to an LTL model that satisfies the same set of LTL formulas. Let us assume that we have a matching logic model $M \vDash \Gamma^{\mathsf{LTL}}$. Our goal is to constructor a model translation function

$$\mathcal{T}^l : \mathbf{Mod}_{\mathsf{ML}}(\Gamma^{\mathsf{LTL}}) \to \mathbf{Mod}_{\mathsf{LTL}}$$

such that $\mathcal{T}^l(M)$ yields the same semantics as M. Recall that an LTL model is an infinite trace over a set of atomic proposition. Therefore, we need to specify the atomic propositions and the infinite trace. Intuitively, we build the following infinite trace $\tau[1], \tau[2], \tau[3], \ldots$, where $\tau[i]$ is the set of atomic propositions that are satisfied in the trace suffix τ, i extracted from M.

First, we need a way to address those elements of a Γ^{LTL}-model that represent trace suffixes.

Definition 10. *For any Γ^{LTL}-model M, we define the function $M_{\bar{\circ}} : M \to M$ defined by $M_{\bar{\circ}}(m) = m'$, where m' is the unique element satisfying $\{m'\} = appext_M(\bar{\circ}_M, \{m\})$, and the function $[\![\_]\!]^{\mathsf{LTL}}_M : \mathbb{N}_{\geq 1} \to [\![TrSuf]\!]_M$ by*

$$[\![n]\!]^{\mathsf{LTL}}_M = \begin{cases} M_{Trace} & \text{if } n = 1 \\ M_{\bar{\circ}}([\![n-1]\!]^{\mathsf{LTL}}_M) & \text{if } n > 1 \end{cases}$$

Intuitively, $[\![i]\!]^{\mathsf{LTL}}_M$ represents the suffix τ, i of the trace τ represented by the model M. With this construction, we can now extract an LTL model out of M.

Definition 11. *Let $\mathcal{T}^l : \mathbf{Mod}_{\mathsf{ML}}(\Gamma^{\mathsf{LTL}}) \to \mathbf{Mod}_{\mathsf{LTL}}$ be the model translation function from matching logic to LTL, defined by $\mathcal{T}^l(M)(i) = \{a \in \mathrm{AP} \mid [\![i]\!]^{\mathsf{LTL}}_M \in a_M\}$ for every $M \in \mathbf{Mod}_{\mathsf{ML}}(\Gamma^{\mathsf{LTL}})$ and $i \in \mathbb{N}_{\geq 1}$.*

The important question is: how does satisfaction in M relate to satisfaction in $\mathcal{T}^l(M)$? The key insight is that from Section 4.3 we already know how does satisfaction in $\mathcal{T}^l(M)$ relate to matching in $\mathcal{M}^l(\mathcal{T}^l(M))$, by instantiating Lemma 8 with $\tau = \mathcal{T}^l(M)$. If we could tie matching in $\mathcal{M}^l(\mathcal{T}^l(M))$ to matching in M, we would get the relationship between matching in M and satisfaction in $\mathcal{T}^l(M)$ by transitivity.

We notice that the set of trace suffixes M_{TrSuf} of a Γ^{LTL}-model M has a structure that is similar to the set of natural numbers: the axiom (TraceSuffix) defining the inhabitant set of the sort *TrSuf* resembles the axiom used for defining natural numbers, with $[\![Trace]\!]$ playing the role of 0 and $\bar{\circ}$ playing the role of the successor function. We also have a function $[\![\_]\!]^{\mathsf{LTL}}_M$ from positive natural numbers to the members of the inhabitant set of the sort *TrSuf*. We now make the relationship precise: we define a function $dist^{\mathsf{LTL}}_M$ going in the other direction and show that the two are inversions of each other. Then we show that $dist^{\mathsf{LTL}}_M$ and $[\![\_]\!]^{\mathsf{LTL}}_M$ preserve matching of LTL formulas, and are therefore exactly the ties between M and $\mathcal{M}^l(\mathcal{T}^l(M))$ that we need.

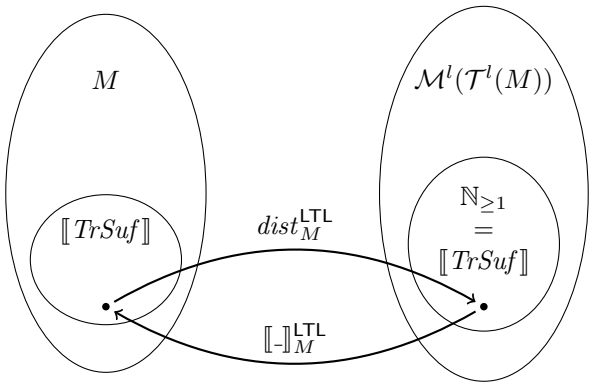


Figure 3: The functions $dist_M^{\mathsf{LTL}}$ and $[\![\_]\!]_M^{\mathsf{LTL}}$ are inversions.

Axioms (Trace) and (TraceSuffix) together with the injectivity of $\bar{\circ}$ (see Lemma 29 in the appendix) ensure that every element $m \in M_{TrSuf}$ can be uniquely constructed by repeatedly applying $\bar{\circ}$ to M_{Trace} - intuitively, every trace suffix can be uniquely constructed by repeatedly incrementing the offset of the trace represented by M. We define the function

$$dist_M^{\mathsf{LTL}} \colon M_{TrSuf} \to \mathbb{N}_{\geq 1}$$

by $dist_M^{\mathsf{LTL}}(m) = n + 1$, where n is the number of times $\bar{\circ}$ needs to be applied to get from M_{Trace} to m. In other words, $dist_M^{\mathsf{LTL}}(m)$ is the $\bar{\circ}$-*distance* of m from M_{Trace}. We call the (unique) sequence of trace suffixes leading from M_{Trace} to m the *initial sequence of* m.

The main component of matching logic models built from LTL models is $[\![TrSuf]\!]_{\mathcal{M}^l(\tau)}$, which is the set of positive natural numbers. The following lemma relates M_{TrSuf} of an arbitrary Γ^{hLTL}-model to $\mathcal{M}^l(\mathcal{T}^l(M))_{TrSuf}$ (see Fig. 3).

Lemma 12. *Functions $[\![\_]\!]_M^{\mathsf{LTL}}$ and $dist_M^{\mathsf{LTL}}$ are inversions of each other, and therefore are bijections between M_{TrSuf} and $\mathcal{M}^l(\mathcal{T}^l(M))_{TrSuf}$.*

The relationship between M and $\mathcal{M}^l(\mathcal{T}^l(M))$ is stronger: the two models are indistinguishable by LTL formulas. We can take a LTL formula, evaluate it in the model M, and then translate the matched elements ("trace suffixes") to elements of $\mathcal{M}^l(\mathcal{T}^l(M))$ - and we get the same result as if we evaluated the formula directly in $\mathcal{M}^l(\mathcal{T}^l(M))$ (see Fig. 4).

Lemma 13. *For every $M \in \mathbf{Mod}_{\mathsf{ML}}(\Gamma^{\mathsf{LTL}})$ and $\varphi \in \mathbf{\Phi}_{\mathsf{LTL}}$,*

$$|\mathsf{L2M}(\varphi)|_{\mathcal{M}^l(\mathcal{T}^l(M))} = dist_M^{\mathsf{LTL}}(|\mathsf{L2M}(\varphi)|_M)\,.$$

And since $dist_M^{\mathsf{LTL}}$ is a bijection, we can do the same in the opposite direction. Therefore, we are able to prove the following counterpart of Lemma 8.

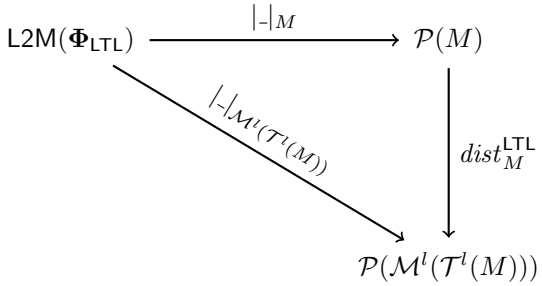


Figure 4: A commutative diagram characterizing $|\text{-}|_{\mathcal{M}^l(\mathcal{T}^l(M))}$.

Lemma 14. *For all* $M \in \mathbf{Mod}_{\mathsf{ML}}(\Gamma^{\mathsf{LTL}})$ *and* $\varphi \in \mathbf{\Phi}_{\mathsf{LTL}}$*,*

$$\mathcal{T}^l(M), i \vDash_{\mathsf{LTL}} \varphi \iff [\![i]\!]_M^{\mathsf{LTL}} \in |\mathsf{L2M}(\varphi)|_M \,.$$

The proof goes as suggested above: we relate the satisfaction in $\mathcal{T}^l(M)$ to matching in $\mathcal{M}^l(\mathcal{T}^l(M))$ using Lemma 8 and then use Lemma 13 and Lemma 12 to bring the reasoning back to M. With Lemma 14, it is now easy to prove the second model equivalence theorem.

Theorem 15 (Model Equivalence B)**.** *For any ML model* $M \in \mathbf{Mod}_{\mathsf{ML}}(\Sigma^{\mathsf{LTL}})$ *and an LTL formula* φ*,*

$$\mathcal{T}^l(M) \vDash_{\mathsf{LTL}} \varphi \iff M \vDash [\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi) \,.$$

4.5 Comparison to [6]

Let us now compare our definition of LTL with the one given in [6], which directly inspired ours. The major distinction is that the axiomatization in [6] allows models that do not correspond to LTL traces, such as a bidirectionally-infinite sequence of elements, multiple identical sequences of elements, or a tree with a root in future. Our axiomatization do not permit such models, and therefore we are able to prove Theorems 9 and 15.

From the technical point of view, one technical distinction between the two formalizations is that in [6], the authors do not explicitly define a formula translation function, because, according to them, matching logic syntax under given signature together with given notations subsume the syntax of LTL. In contrast, we feel forced to give an explicit definition of the formula translation function because of the negation operator: since the authors of [6] use a sorted variant of matching logic where the built-in negation preserves sorts, the LTL negation operator can be, in [6], (implicitly) directly mapped to the negation operator of matching logic; on the other hand, we need to explicitly "filter" the result of matching logic negation, as explained above. Otherwise, the syntactic part is similar, except that in [6], the symbol in signature is called $\bullet$, and the LTL operator $\circ$ is defined as a notation $\circ X \equiv \neg\bullet\neg X$.

The second technical distinction is that in [6], two axioms are sufficient to capture the semantics:

$$(\textsc{Inf}) \quad \bullet\top \qquad\qquad (\textsc{Lin}) \quad \bullet X \to \circ X$$

Intuitively, their axiom (Inf) corresponds to our axiom (Inf), and its simplicity is due to the many-sorted nature of many-sorted matching logic: $\top$ in their axiom evalutes to all elements of the sort that they (implicitly) use, and their axiom (Inf) therefore corresponds to our pattern $[\![\mathit{TrSuf}]\!] = \circ[\![\mathit{TrSuf}]\!]$, of which, so to speak, our axiom (Inf) is a half. We enforce the other inclusion using the axiom (NextPFun). The other axiom of [6], (Lin), prevents the model elements from branching in the direction of $\circ$, that is, it enforces *linear future*. Our axioms (NextPFun) and (NextInj) enforce not only linear future, but also linear past.

The third technical distinction is in the way we formulate our arguments. Our argument is purely model-theoretical, while the argument of [6] is based on the observation that their models form a class of transition systems, and on the completeness of modal μ-logic, which is a fragment of matching logic.

5 Defining HyperLTL in Matching Logic

In this section we review HyperLTL, define it as a matching logic specification HLTL, define translations from HyperLTL to matching logic models and back, and show that our definitions are correct by proving model equivalence theorems (Theorems 2 and 3). We use similar ideas as for LTL in Section 4.

5.1 HyperLTL Syntax and Semantics

HyperLTL is an extension of LTL for specifying *hyperproperties*, i.e., properties of *sets of traces* instead of properties about traces. The *syntax* of HyperLTL is parametric in

- a countable set AP of *atomic propositions* denoted as a;
- an countably infinite set V of *path variables* denoted as $\pi, \pi_1, \pi_2, \ldots$.

HyperLTL extends LTL with quantifiers over path variables, which are then used to label atomic propositions; quantifiers appear only at the top level:

$$\begin{aligned} \varphi &::= \forall \pi \,.\, \varphi \mid \exists \pi \,.\, \varphi \mid \psi \\ \psi &::= a_\pi \mid \neg\psi \mid \psi \vee \psi \mid \circ\psi \mid \psi \,U\, \psi \end{aligned}$$

We let $\mathbf{\Phi}_{\mathsf{hLTL}}$ denote the set of all HyperLTL formulas defined by the above grammar.

The semantics of HyperLTL is defined w.r.t. a nonempty set of traces T, called a *HyperLTL model*, a valuation $\Pi\colon V \to T$, and a number $i \in \mathbb{N}_{\geq 1}$, as the relation $\vDash_{\mathsf{hLTL}} \subseteq \mathbf{Mod}_{\mathsf{hLTL}} \times (V \to T) \times \mathbb{N}_{\geq 1} \times \mathbf{\Phi}_{\mathsf{hLTL}}$ inductively defined as follows:

- $T, \Pi, i \vDash_{\mathsf{hLTL}} a_\pi$ iff $a \in \Pi(\pi)[i]$;
- $T, \Pi, i \vDash_{\mathsf{hLTL}} \exists \pi \,.\, \varphi$ iff there exists $\tau \in T$ such that $T, \Pi[\tau/\pi], i \vDash_{\mathsf{hLTL}} \varphi$;
- $T, \Pi, i \vDash_{\mathsf{hLTL}} \forall \pi \,.\, \varphi$ iff for all $\tau \in T$ we have $T, \Pi[\tau/\pi], i \vDash_{\mathsf{hLTL}} \varphi$;

where $\Pi[\tau/\pi]$ denotes the valuation Π' such that $\Pi'(\pi) = \tau$ and $\Pi'(\pi') = \Pi(\pi)$ for all $\pi' \not\equiv \pi$. (For brevity, we omit the definitions for $\neg$, $\vee$, $\circ$, and U, since they are similar to LTL.) We write $T \vDash_{\mathsf{hLTL}} \varphi$ if $T, \Pi, 1 \vDash_{\mathsf{hLTL}} \varphi$ for all valuations Π. We write $\vDash_{\mathsf{hLTL}} \varphi$ if $T \vDash_{\mathsf{hLTL}} \varphi$ for all T. We let $\mathbf{Mod}_{\mathsf{hLTL}} = \mathcal{P}((\mathcal{P}(\mathrm{AP}))^\omega)$ denote the set of all HyperLTL models.

5.2 Capturing HyperLTL in Matching Logic

For HyperLTL, we let $V \subseteq \mathrm{EV}$; i.e., the set of matching logic element variables contains all HyperLTL path variables. The matching logic specification that defines the theory Γ^{hLTL} is shown in Spec. 7. The specification includes all LTL axioms with one modification: the axiom (Trace) now enforces the existence of a trace, but does not limit their number. The axioms (Row), (SC), (Col), and (Eq) define the meaning of the symbols *row*, *sc*, *col*, and *eq*, respectively: given trace suffixes x, y, *row* x represents the set of all trace suffixes that are forward-reachable (using $\bar{\circ}$) or backward-reachable (using $\circ$) from x, which is illustrated as a row on Fig. 5; *sc* x y is a predicate which holds if x and y are at the same distance from their (complete) trace, which in terms of Fig. 5 means they are at the same column; *col* x represents the set of trace suffixes that are at the same distance from their (complete) trace, which is illustrated as a column; and the predicate *eq* x y holds if x and y validate the same set of atomic proposition from that point forward. The axiom (Set) guarantees that the model does not contain two equivalent traces; i.e., that traces form a set (and not a general multiset), as in HyperLTL models.

As in the LTL case, HyperLTL formulas maps directly to the syntax of matching logic and the defined syntactic sugar, except that in the case of quantifiers we need to restrict the quantification to full traces.

Definition 16. *We define the function* $\mathsf{H2M}(\_) : \mathbf{\Phi}_{\mathsf{hLTL}} \to \textsc{Pattern}(\Sigma^{\mathsf{hLTL}})$ *inductively by*

- $\mathsf{H2M}(a_\pi) = a_\pi$*;*
- $\mathsf{H2M}(\neg\psi) = \neg_{\mathsf{hLTL}} \mathsf{H2M}(\psi)$*;*
- $\mathsf{H2M}(\psi_1 \vee \psi_2) = \mathsf{H2M}(\psi_1) \vee \mathsf{H2M}(\psi_2)$*;*
- $\mathsf{H2M}(\circ\psi) = \circ \mathsf{H2M}(\psi)$*;*
- $\mathsf{H2M}(\psi_1 \, U \, \psi_2) = \mathsf{H2M}(\psi_1) \, U \, \mathsf{H2M}(\psi_2)$*;*
- $\mathsf{H2M}(\forall \pi . \varphi) = \forall \pi \colon \mathit{Trace} . \mathsf{H2M}(\varphi)$*; and*
- $\mathsf{H2M}(\exists \pi . \varphi) = \exists \pi \colon \mathit{Trace} . \mathsf{H2M}(\varphi)$.

spec HLTL
Import: SORTS, RECURSION
Symbol: Trace, TrSuf, $\circ$, $\bar{\circ}$,
atomic proposition a *(for every* $a \in \mathrm{AP}$*)*,
row, col, sc, eq
Notation:
$\neg_{\mathsf{hLTL}}\varphi \equiv [\![\mathit{TrSuf}]\!] \wedge \neg\varphi$
$\varphi_1 \, U \, \varphi_2 \equiv \mu X \,.\, \varphi_2 \vee (\varphi_1 \wedge \circ X)$
$a_\pi \equiv \mathit{col}(a \wedge \mathit{row}(\pi))$
Axiom:

(Prev)	$\bar{\circ}x = \exists y \,.\, y \wedge (x \in \circ y))$
(Trace)	$\exists x.\, x \in [\![\mathit{Trace}]\!]$
(TraceSuffix)	$[\![\mathit{TrSuf}]\!] = \mu X.\, [\![\mathit{Trace}]\!] \vee \bar{\circ} X$
(Inf)	$[\![\mathit{TrSuf}]\!] \subseteq \circ [\![\mathit{TrSuf}]\!]$
(NextOut)	$\circ(\neg [\![\mathit{TrSuf}]\!]) \subseteq \neg [\![\mathit{TrSuf}]\!]$
(NextPFun)	$\circ\colon \mathit{TrSuf} \rightharpoonup \mathit{TrSuf}$
(NextInj)	$\forall x_1, x_2\colon \mathit{TrSuf}.\, \circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot$ $\to x_1 = x_2$
(AtomicProp)	$a \subseteq [\![\mathit{TrSuf}]\!]$
(Row)	$\forall x\colon \mathit{TrSuf}.\, \mathit{row}\ x = \mu X \,.\, x \vee \circ X \vee \bar{\circ} X$
(SC)	$\forall x, y\colon \mathit{TrSuf}.\, \mathit{sc}\ x\ y$ $=_{\mathrm{lfp}} (x \in [\![\mathit{Trace}]\!] \wedge y \in [\![\mathit{Trace}]\!])$ $\vee \mathit{sc}\ (\circ x)\ (\circ y)$
(Col)	$\forall x\colon \mathit{TrSuf}.\, \mathit{col}\ x$ $= \exists y : \mathit{TrSuf}.\, y \wedge (\mathit{sc}\ x\ y)$
(Eq)	$\forall x, y\colon \mathit{TrSuf}.\, \mathit{eq}\ x\ y$ $=_{\mathrm{gfp}} (\bigwedge_{a \in \mathrm{AP}} x \in a \leftrightarrow y \in a)$ $\wedge \mathit{eq}\ (\bar{\circ} x)\ (\bar{\circ} y)$
(Set)	$\forall x, y\colon \mathit{Trace}.\, \mathit{eq}\ x\ y \to x = y$

endspec

Spec. 7: HyperLTL as a matching logic specification

5.3 From HyperLTL Models to Γ^{hLTL}-Models

In this section we define a function translating HyperLTL models into matching logic models and prove the first model equivalence theorem, Theorem 2. We formally define a model translation function $\mathcal{M}^h : \mathbf{Mod}_{\mathsf{hLTL}} \to \mathbf{Mod}_{\mathsf{ML}}(\Gamma^{\mathsf{hLTL}})$ in Definition 39 in the appendix; here we only intuitively illustrate its basic properties. For any HyperLTL model (i.e., a nonempty set of traces) T, the carrier set of the translated model $\mathcal{M}^h(T)$ contains the original HyperLTL traces paired with positive natural numbers: $\mathcal{M}^h(T)_{\mathit{TrSuf}} = T \times \mathbb{N}_{\geq 1}$; each number representing an offset on a trace. The symbol $\circ$ is interpreted as a partial function that decreases the offset if it is greater than 1, while the symbol $\bar{\circ}$ is interpreted as a function that increases the offset. This is illustrated in Fig. 5, where the filled

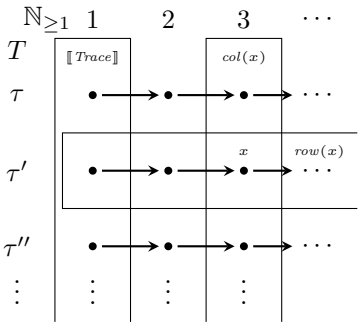


Figure 5: An intuitive illustration of the Γ^{hLTL}-model generated from the HyperLTL model T.

---

circles represent elements of $\mathcal{M}^h(T)_{TrSuf}$ and the arrows represent the function that interprets $\bar{\mathsf{o}}$.

We also need to construct matching logic valuations from HyperLTL ones. Recall that a HyperLTL valuation $\Pi\colon V \to T$ maps HyperLTL variables to traces. The corresponding matching logic valuation $\rho_T^{\mathsf{hLTL}}(\Pi)$ needs to map a HyperLTL variable π to some pair in $T \times \mathbb{N}_{\geq 1}$. A natural choice is to define $\rho_T^{\mathsf{hLTL}}(\Pi)(\pi) = (\Pi(\pi), 1)$, since the number 1 represents the smallest possible offset on the trace $\Pi(\pi)$, and since the resulting value is an inhabitant of the sort *Trace*.

The reader might wonder why the notation for a_π is defined the way it is. Intuitively, this is because in HyperLTL semantics, all traces in a model are considered from a certain point in time, which is the same for all traces; this point is represented by the $\mathbb{N}_{\geq 1}$ component of the relation $\vDash_{\mathsf{hLTL}}$. Therefore, in matching logic, we too need to consider all traces from a certain point at once. Our idea is that in matching logic, HyperLTL formulas match not only individual trace suffixes, but whole *columns* (in the sense of Fig. 5) of trace suffixes, where columns represents the idea of "traces from a certain (the same) point". Given any column, a HyperLTL formula either matches the full column, or none of it. This is easily seen when we consider the base-case HyperLTL formula a_π (which in matching logic desugars to $col(a \wedge row(\pi))$): if the matching logic formula a matches a trace suffix x that lies in the row corresponding to the HyperLTL variable π , then the matching logic formula $a \wedge row(\pi)$ still matches x, and $col(a \wedge row(\pi))$ matches the whole column in which x lies. Consequently, if a_π matches some part of a column, it matches the column as a whole. In the following lemma that connect the semantics of HyperLTL with semantics of matching logic we therefore have two cases: one where the HyperLTL formula matches the whole column, and the other where it matches none of it.

Lemma 17. *For any HyperLTL model* $T \in \mathbf{Mod}_{\mathsf{hLTL}}$*, valuation* $\Pi : V \to T$*,*

and $i \in \mathbb{N}_{\geq 1}$,

$$T, \Pi, i \vDash_{\mathsf{hLTL}} \varphi \iff [\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\varphi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}, \text{ and}$$
$$T, \Pi, i \nvDash_{\mathsf{hLTL}} \varphi \iff [\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\mathsf{H2M}(\varphi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \emptyset\,,$$

where $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)}$ *stands for the i-th column of the model* $\mathcal{M}^h(T)$*; that is, for the set* $\{(\tau, i) \mid \tau \in T\}$.

Now we would like to prove the first model equivalence theorem, as we did in the LTL case. However, there is one catch: the implication "$T \vDash_{\mathsf{hLTL}} \varphi$ implies $\mathcal{M}^h(T) \vDash [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)$" does not hold in general, when φ is not closed. Consider the HyperLTL model $T = \{\tau\}$ consisting of a single trace, $\tau = \{a\}, \emptyset, \emptyset, \ldots$, and the HyperLTL formula $\varphi \equiv \exists \pi'.\, G(a_\pi \leftrightarrow a_{\pi'})$, where $G(\psi) \equiv \neg(\mathit{true}\; U\; \neg\psi)$. Then $T \vDash_{\mathsf{hLTL}} \varphi$ holds (in fact, φ is a HyperLTL tautology), but $\mathcal{M}^h(T) \nvDash [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)$, since $(\tau, 1) \notin |\mathsf{H2M}(\varphi)|_\rho$ for the $\mathcal{M}^h(T)$-valuation ρ such that $\rho(\pi) = (\tau, 2)$. The problem is that ρ does not correspond to any HyperLTL valuation, since it maps π outside $[\![\mathit{Trace}]\!]$. To fix this, we define the formula $\mathsf{WellSorted}(\varphi)$ that is true only if the valuation maps all free variables to the inhabitants of the sort *Trace*:

$$\mathsf{WellSorted}(\varphi) \equiv \bigwedge_{\pi \in FV(\mathsf{H2M}(\varphi))} \pi \in [\![\mathit{Trace}]\!]\,.$$

Now we are ready to prove the first model equivalence theorem.

Theorem 2 (Model Equivalence A)**.** *For any HyperLTL model* T *and formula* φ*,*

$$T \vDash_{\mathsf{hLTL}} \varphi \text{ iff } \mathcal{M}^h(T) \vDash \mathit{WellSorted}(\varphi) \rightarrow [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)$$

5.4 From Γ^{hLTL}-Models to HyperLTL Models

In this section we define a translation function from matching logic models satisfying Γ^{hLTL} to HyperLTL models, and prove the second model equivalence theorem. We use the same technique as in the LTL case; namely, we define the translation function in terms of $\bar{\circ}$-distance and *initial sequences*, and prove the theorem by establishing a relationship between arbitrary Γ^{hLTL}-models and matching logic models constructed from HyperLTL models.

5.4.1 Constructing a HyperLTL Model

Recall that a HyperLTL model is a set of traces, and that we represent traces as inhabitants of the sort *Trace*. Therefore, given a Γ^{hLTL}-model M, we define the HyperLTL model

$$\mathcal{T}^h(M) = \{\tau(m) \mid m \in |[\![\mathit{Trace}]\!]|_M\}\,.$$

In other words, the model contains one trace $\tau(m)$ for each inhabitant of the sort *Trace*, where the trace $\tau(m)$ is defined similarly as in the LTL case:

$$\tau(m)[i] = \{a \mid \mathit{get}(m, i) \in a_M\}\,,$$

where *get* is an iterative application of $\bar{\circ}$: $get(m,1) = m$ and $get(m,n+1) = M_{\bar{\circ}}(get(m,n))$.

5.4.2 Bijection β_M

There exists a bijection β_M between M and $\mathcal{M}^h(\mathcal{T}^h(M))$. In the case of LTL, we had a function $dist_M^{\mathsf{LTL}}$ that served as a matching-preserving bijection from an arbitrary matching logic Γ^{LTL}-model M to the constructed model $\mathcal{M}^l(\mathcal{T}^l(M))$. Recall that $dist_M^{\mathsf{LTL}}$ calculated the $\bar{\circ}$-distance from the full trace to the given trace suffix, and returned a value in $\mathbb{N}_{\geq 1}$, which, conveniently, was also the inhabitant set of *TrSuf* in the constructed matching logic model. In the case of HyperLTL we define a similar function, denoted $dist_M^{\mathsf{hLTL}}$, with $\mathbb{N}_{\geq 1}$ as codomain. Since $\mathbb{N}_{\geq 1}$ is now only one component of the inhabitant set of *TrSuf*, that is, of $T \times \mathbb{N}_{\geq 1}$, we also need a function $full_M^{\mathsf{hLTL}}$ that compute a HyperLTL trace $\tau \in T$ from the given trace suffix of M. These two functions, when paired, then form a matching-preserving bijection β between M and $\mathcal{M}^h(\mathcal{T}^h(M))$.

We define the function $full_M^{\mathsf{hLTL}} \colon M_{TrSuf} \to M_{Trace}$ such that $full_M^{\mathsf{hLTL}}(m)$ is the (unique) element of M_{Trace} from which m is $\bar{\circ}$-reachable (i.e., the *full trace* of the trace suffix m), and the function $dist_M^{\mathsf{hLTL}} \colon M_{TrSuf} \to \mathbb{N}_{\geq 1}$ by $dist_M^{\mathsf{hLTL}}(m) = n{+}1$, where n is the number of times $\bar{\circ}$ needs to be applied to get from $full_M^{\mathsf{hLTL}}(m)$ to m.

Now we define the matching preserving bijection β by pairing $full_M^{\mathsf{hLTL}}$ and $dist_M^{\mathsf{hLTL}}$. Intuitively, β decomposes elements representing trace suffixes in model M into a (full) trace and an offset on the trace.

Definition 18. *For any Γ^{hLTL}-model M, we define the function $\beta_M : M \to \mathcal{M}^h(\mathcal{T}^h(M)) = \mathcal{T}^h(M) \times \mathbb{N}$ by*

$$\beta_M(m) = (\tau(full_M^{\mathsf{hLTL}}(m)), dist_M^{\mathsf{hLTL}}(m)) .$$

5.4.3 β_M Preserves Semantics

Now we want to prove Lemma 20, which intuitively says that if we interpret a formula φ in some model M and then use β_M to transfer the interpretation to $\mathcal{M}^h(\mathcal{T}^h(M))$, we get the same result as is we interpreted φ directly in $\mathcal{M}^h(\mathcal{T}^h(M))$. In the case of LTL, the function $dist_M^{\mathsf{LTL}}$ preserved semantics irrespectively of valuations, since LTL formulas have no variables. The HyperLTL case is more complicated because of variables. But just as there exist Σ^{hLTL}-models that do not correspond to any HyperLTL model, and we filter them out using the theory Γ^{hLTL}, for any Γ^{hLTL}-model $\mathcal{M}(T)$ built of a HyperLTL model T there are some M-valuations that do not correspond to any HyperLTL T-valuation; we are not interested in those. The valuations we are interested in, we call "well-sorted".

Definition 19. *A valuation $\rho : \textsc{Var} \to M$ is called "well-sorted" if $\rho(\pi) \in M_{Trace}$ for any path variable $\pi \in V$.*

In other words, in a well-sorted valuation all path variables range over the sort *Trace*, as expected. A valuation $\rho_T^{\mathsf{hLTL}}(\Pi)$ is always well-sorted. Moreover, for any well-sorted $\mathcal{M}^h(T)$-valuation ρ there exists a HyperLTL T-valuation Π' such that ρ is equivalent to $\rho_T^{\mathsf{hLTL}}(\Pi')$. Similarly to the LTL case, HyperLTL formulas in well-sorted matching logic valuations evaluate to trace suffixes. Now we can formulate the matching-preserving lemma.

Lemma 20. *For every Γ^{hLTL}-model M, every well-sorted M-valuation ρ, and every HyperLTL formula φ,*

$$|\mathsf{H2M}(\varphi)|_{\rho_{\Pi(\rho)}} = \beta_M(|\mathsf{H2M}(\varphi)|_\rho)$$

Similarly to the LTL case, this lemma is useful in that it allows us to transfer HyperLTL reasoning between the model M and the model $\mathcal{M}^h(\mathcal{T}^h(M))$.

5.4.4 Model Equivalence

Now we are going to prove the second model equivalence theorem (Theorem 3) using a lemma that connects the semantics in M with semantics in $\mathcal{T}^h(M)$ (Lemma 21). We just need two more technical details before doing so: (1) to transform well-sorted matching logic valuations into HyperLTL valuations, and (2) to denote columns of trace suffixes. For (1), given a matching logic valuation $\rho : V \to M$, we define the corresponding HyperLTL valuation $\Pi(\rho) : V \to \mathcal{T}^h(M)$ as $\Pi(\rho)(\pi) = \tau(\mathit{full}_M^{\mathsf{hLTL}}(\rho(\pi)))$. This way, if we create a matching logic valuation from a HyperLTL valuation and then turn it back into a HyperLTL valuation, we get back the original one. For (2), we want $[\![i]\!]_M^{\mathsf{hLTL}}$ to stand for the i-th column of the model M - if there is something like columns in an arbitrary Γ^{hLTL}-model. We can define $[\![i]\!]_M^{\mathsf{hLTL}}$ using repeated applications of $\bar{\circ}$ to *Trace*. Such definition coincides with the notation we have seen in Lemma 17. Now we can formulate the following lemma, which is a counterpart of Lemma 17 in the same sense in which Lemma 14 is a counterpart of Lemma 8.

Lemma 21. *For every model M of Γ^{hLTL}, every HyperLTL formula φ, and every well-sorted valuation $\rho : V \to M$,*

$$\begin{aligned} T, \Pi(\rho), i \vDash_{\mathsf{hLTL}} \mathsf{H2M}(\varphi) &\iff [\![i]\!]_M^{\mathsf{hLTL}} \subseteq |\mathsf{H2M}(\varphi)|_\rho \\ T, \Pi(\rho), i \vDash_{\mathsf{hLTL}} \mathsf{H2M}(\varphi) &\iff [\![i]\!]_M^{\mathsf{hLTL}} \cap |\mathsf{H2M}(\varphi)|_\rho \neq \emptyset\,, \end{aligned}$$

where $T = \mathcal{T}^h(M)$.

The lemma is proved similarly to Lemma 14, using Lemma 17 and properties of β_M. With this lemma it is easy to prove the second model equivalence theorem, which we restate here.

Theorem 3 (Model Equivalence B)**.** *For any matching logic model $M \vDash \Gamma^{\mathsf{hLTL}}$ and HyperLTL formula φ,*

$$\mathcal{T}^h(M) \vDash_{\mathsf{hLTL}} \varphi \text{ iff } M \vDash \mathit{WellSorted}(\varphi) \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)$$

To summarize: the theory Γ^{hLTL} captures HyperLTL models, which we have shown by defining functions $\mathcal{M}^h$ and $\mathcal{T}^h$ for translation HyperLTL models to matching logic models and back and proving model equivalence theorems (Theorems 2 and 3). In the next section we discuss how to apply our results for HyperLTL validity and and model checking.

6 Applications

The model equivalence theorems (Theorems 9, 15, 2, 3) have two important applications: (1) they allow us to check whether a (Hyper)LTL formula is a tautology, by easily proving a semantic equivalence theorem; and (2) they enable us to do model checking of a (Hyper)LTL formula against a (Hyper)LTL model specified in matching logic.

6.1 Validity Checking

Validity checking, that is, checking whether a formula is true in all models, is enabled by a semantic equivalence theorem, which is a direct consequence of model equivalence theorems. For LTL, we can state:

Theorem 22 (Semantic Equivalence)**.** *For any* $\varphi \in \mathbf{\Phi}_{\mathsf{LTL}}$*,*

$$\vDash_{\mathsf{LTL}} \varphi \iff \Gamma^{\mathsf{LTL}} \vDash [\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi)$$

The implication from left to right follows from Theorem 15, while the other implication from Theorem 9. Using Theorems 3 and 2, for HyperLTL we can prove a similar result, that is, Theorem 1, which we restate here:

Theorem 1 (Semantic Equivalence)**.** *Let* φ *be any HyperLTL formula and* $\mathsf{H2M}(\varphi)$ *its translation in matching logic. Then:*

$$\vDash_{\mathsf{hLTL}} \varphi \textit{ iff } \Gamma^{\mathsf{hLTL}} \vDash \mathit{WellSorted}(\varphi) \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)$$

With these semantic equivalence theorems we can reduce (Hyper)LTL validity to matching logic validity.

6.2 Model Checking

The model equivalence theorems (Theorems 2 and 3) enable us to do model checking, that is, to check whether a HyperLTL model satisfies a HyperLTL formula. For example, let us suppose that, given the signature $\mathrm{AP} = \{\mathit{red}\}$, we want to check whether the HyperLTL model $T = \{\tau_a, \tau_b, \tau_c\}$, where $\tau_a = \emptyset, \emptyset, \ldots$, $\tau_b = \{\mathit{red}\}, \emptyset, \emptyset, \ldots$, and $\tau_c = \{\mathit{red}\}, \emptyset, \{\mathit{red}\}, \emptyset, \ldots$, satisfies the HyperLTL formula φ. We can define a the theory Γ^T consistent with Γ^{hLTL} such that every matching logic model M of $\Gamma^T \cup \Gamma^{\mathsf{hLTL}}$ translates to T (meaning that $\mathcal{T}^h(M) = T$). Then, using Theorem 3, it follows that $T \vDash_{\mathsf{hLTL}} \varphi$ iff

$$\Gamma^T \cup \Gamma^{\mathsf{hLTL}} \vDash \mathsf{WellSorted}(\varphi) \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)\,.$$

The consistency of Γ^T with Γ^{hLTL} and the uniqueness of a HyperLTL model are the only assumptions here. In particular, the axioms of Γ^T are not limited to translated HyperLTL formulas, but they can use the full power of matching logic. For example, Γ^T can be defined as in Spec. 8.

```
spec EXAMPLE
  Import: HLTL , NAT , PAIR
  Symbol: a, b, c
  Axiom:
    (TraceT)    [[Trace]] = pair (a ∨ b ∨ c) zero
    (TrSufT)    [[TrSuf]] = pair (a ∨ b ∨ c)([[Nat]])
    (redT)      red = (pair b zero)∨
                pair c (μX. zero ∨ succ (succ X))
endspec
```

Spec. 8: A specification of the example HyperLTL model T.

We conjecture that in this way, one could model-check Kripke structures against a HyperLTL specification: we could define a *generator* theory Γ^{gen} that generates a set of traces from a Kripke structure. The generator theory would need to have the property that for any theory Γ^{KS} that uniquely represents a Kripke structure, the theory $\Gamma^{KS} \cup \Gamma^{gen} \cup \Gamma^{\mathsf{hLTL}}$ is consistent and all its models represent the same HyperLTL model. We leave the development of Γ^{gen} as a future work.

6.3 Formal Reasoning for HyperLTL

As noted in Section 2, matching logic has a proof system that is sound for all theories. Therefore, we obtain a formal reasoning system for HyperLTL for free: one can show the validity of the matching logic pattern

$$\mathsf{WellSorted}(\varphi) \to [\![Trace]\!] \subseteq \mathsf{H2M}(\varphi)$$

in the respective theory (Γ^{hLTL}, or $\Gamma^T \cup \Gamma^{\mathsf{hLTL}}$, or $\Gamma^{KS} \cup \Gamma^{gen} \cup \Gamma^{\mathsf{hLTL}}$, depending whether the goals is to validity-check or model-check) by constructing a formal proof in the proof system. We leave the completeness of the proof system in these theories for future work.

7 Related Work

The semantic equivalence theorem for LTL (Theorem 22) has been already proved in [6] for an axiomatization that is simpler then ours; however, that axiomatization permits matching logic models for which no equivalent LTL model exists, and thus does not satisfy the model equivalence theorems. Theorems 9 and 15 are therefore our new contribution.

Both matching logic and hyperproperties are active research areas. On the matching logic side, some work has been done on automated reasoning [8]; on defining other logical systems inside matching logic, such as initial algebras [5], hybrid automata [28] and type systems [7]; and on connecting matching logic to other logical frameworks, such as rewriting logic and constrained constructor patterns [4], and hybrid modal logic [21]. On the hyperproperties side, various logics have been proposed, such as HyperCTL* [10], PHL (Probabilistic Hyper Logic) [12], HyperPDL-Δ (Propositional Dynamic Logic for Hyperproperties) [18] and HyperMTL [2]; and model checking [2, 16], runtime verification and monitoring [14, 15], and synthesis algorithms [13] have been investigated.

8 Future Work and Conclusion

We defined a matching logic theory Γ^{hLTL} of HyperLTL and proved a semantic equivalence theorem that enables us to check validity of a HyperLTL formula inside matching logic, and a model equivalence theorem that allows model checking HyperLTL formulas against a HyperLTL model inside matching logic. The theory is closely related to our axiomatization Γ^{LTL} of LTL in matching logic, for which we proved similar results. We leave a practical application of our result to model checking systems against HyperLTL properties in a matching logic prover for a future work. In future, we also intend to define a matching logic theory of HyperCTL*, a logic that is an extension of HyperLTL and CTL*, and study Cartesian Hoare logic in matching logic.

References

[1] Denis Bogdanas and Grigore Rosu. K-java: A complete semantics of java. In Sriram K. Rajamani and David Walker, editors, *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015*, pages 445–456. ACM, 2015.

[2] Borzoo Bonakdarpour, Pavithra Prabhakar, and César Sánchez. Model checking timed hyperproperties in discrete-time systems. In Ritchie Lee, Susmit Jha, and Anastasia Mavridou, editors, *NASA Formal Methods - 12th International Symposium, NFM 2020, Moffett Field, CA, USA, May 11-15, 2020, Proceedings*, volume 12229 of *Lecture Notes in Computer Science*, pages 311–328. Springer, 2020.

[3] Borzoo Bonakdarpour, César Sánchez, and Gerardo Schneider. Monitoring hyperproperties by combining static analysis and runtime verification. In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation. Verification - 8th International Symposium, ISoLA 2018, Limassol, Cyprus, November 5-9, 2018,*

Proceedings, Part II, volume 11245 of *Lecture Notes in Computer Science*, pages 8–27. Springer, 2018.

[4] Xiaohong Chen, Dorel Lucanu, and Grigore Roşu. Connecting constrained constructor patterns and matching logic. In *Proceedings of the 13th International Workshop on Rewriting Logic and Its Applications (WRLA'20)*, April 2020.

[5] Xiaohong Chen, Dorel Lucanu, and Grigore Rosu. Initial algebra semantics in matching logic. Technical report, 2020.

[6] Xiaohong Chen and Grigore Rosu. Matching μ-logic. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–13. IEEE, 2019.

[7] Xiaohong Chen and Grigore Rosu. A general approach to define binders using matching logic. *Proc. ACM Program. Lang.*, 4(ICFP):88:1–88:32, 2020.

[8] Xiaohong Chen, Minh-Thai Trinh, Nishant Rodrigues, Lucas Peña, and Grigore Roşu. Towards a unified proof framework for automated fixpoint reasoning using matching logic. In *PACMPL Issue OOPSLA 2020*, pages 1–29. ACM/IEEE, Nov 2020.

[9] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In Dexter Kozen, editor, *Logics of Programs, Workshop, Yorktown Heights, New York, USA, May 1981*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer, 1981.

[10] Michael R. Clarkson, Bernd Finkbeiner, Masoud Koleini, Kristopher K. Micinski, Markus N. Rabe, and César Sánchez. Temporal logics for hyperproperties. In Martín Abadi and Steve Kremer, editors, *Principles of Security and Trust - Third International Conference, POST 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8414 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2014.

[11] Michael R. Clarkson and Fred B. Schneider. Hyperproperties. *J. Comput. Secur.*, 18(6):1157–1210, 2010.

[12] Rayna Dimitrova, Bernd Finkbeiner, and Hazem Torfah. Probabilistic hyperproperties of markov decision processes. In Dang Van Hung and Oleg Sokolsky, editors, *Automated Technology for Verification and Analysis - 18th International Symposium, ATVA 2020, Hanoi, Vietnam, October 19-23, 2020, Proceedings*, volume 12302 of *Lecture Notes in Computer Science*, pages 484–500. Springer, 2020.

[13] Bernd Finkbeiner, Christopher Hahn, Philip Lukert, Marvin Stenger, and Leander Tentrup. Synthesis from hyperproperties. *Acta Informatica*, 57(1-2):137–163, 2020.

[14] Bernd Finkbeiner, Christopher Hahn, Marvin Stenger, and Leander Tentrup. Rvhyper: A runtime verification tool for temporal hyperproperties. In Dirk Beyer and Marieke Huisman, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 24th International Conference, TACAS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings, Part II*, volume 10806 of *Lecture Notes in Computer Science*, pages 194–200. Springer, 2018.

[15] Bernd Finkbeiner, Christopher Hahn, Marvin Stenger, and Leander Tentrup. Monitoring hyperproperties. *Formal Methods Syst. Des.*, 54(3):336–363, 2019.

[16] Bernd Finkbeiner, Markus N. Rabe, and César Sánchez. Algorithms for model checking hyperltl and hyperctl ^*. In Daniel Kroening and Corina S. Pasareanu, editors, *Computer Aided Verification - 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part I*, volume 9206 of *Lecture Notes in Computer Science*, pages 30–48. Springer, 2015.

[17] Joseph A. Goguen and José Meseguer. Security policies and security models. In *1982 IEEE Symposium on Security and Privacy, Oakland, CA, USA, April 26-28, 1982*, pages 11–20. IEEE Computer Society, 1982.

[18] Jens Oliver Gutsfeld, Markus Müller-Olm, and Christoph Ohrem. Propositional dynamic logic for hyperproperties. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPIcs*, pages 50:1–50:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[19] Chris Hathhorn, Chucky Ellison, and Grigore Rosu. Defining the undefinedness of C. In David Grove and Steve Blackburn, editors, *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation, Portland, OR, USA, June 15-17, 2015*, pages 336–345. ACM, 2015.

[20] Dexter Kozen. Results on the propositional μ-calculus. In Mogens Nielsen and Erik Meineche Schmidt, editors, *Automata, Languages and Programming*, pages 348–359, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg.

[21] Ioana Leucstean, Natalia Moangba, and Traian Florin cSerbbanuctba. From hybrid modal logic to matching logic and back. 2019.

[22] Corto Mascle and Martin Zimmermann. The keys to decidable hyperltl satisfiability: Small models or very simple formulas. In Maribel Fernández and Anca Muscholl, editors, *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain*, volume 152 of *LIPIcs*, pages 29:1–29:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[23] Daejun Park, Andrei Stefanescu, and Grigore Rosu. KJS: a complete formal semantics of javascript. In David Grove and Steve Blackburn, editors, *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation, Portland, OR, USA, June 15-17, 2015*, pages 346–356. ACM, 2015.

[24] Srinivas Pinisetty, Thibaud Antignac, David Sands, and Gerardo Schneider. Monitoring data minimisation. *CoRR*, abs/1801.02484, 2018.

[25] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 46–57. IEEE Computer Society, 1977.

[26] Amir Pnueli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, SFCS '77, page 46–57, USA, 1977. IEEE Computer Society.

[27] Grigore Rosu. Matching logic. *Log. Methods Comput. Sci.*, 13(4), 2017.

[28] Manasvi Saxena, Nishant Rodrigues, Xiaohong Chen, and Grigore Roşu. Formal semantics of hybrid automata. Technical Report http://hdl.handle.net/2142/106822, University of Illinois at Urbana-Champaign, April 2020.

[29] Marcelo Sousa and Isil Dillig. Cartesian hoare logic for verifying k-safety properties. In Chandra Krintz and Emery Berger, editors, *Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2016, Santa Barbara, CA, USA, June 13-17, 2016*, pages 57–69. ACM, 2016.

[30] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.*, 5(2):285–309, 1955.

A Proofs

A.1 ML

Lemma 23. *Let M be a set and $F_1, F_2 \colon \mathcal{P}(M) \to \mathcal{P}(M)$ monotonic function such that for any $A \subset M$, $F_1(A) \subseteq F_2(A)$. Then $\mu F_1 \subseteq \mu F_2$.*

Proof. Since μF_1 is the least prefixpoint of F_1, it is enough to sho that μF_2 is a prefixpoint of F_1; i.e., that $F_1(\mu F_2) \subseteq \mu F_2$. But that holds by $F_1(\mu F_2) \subseteq F_2(\mu F_2) = \mu F_2$. □

Definition 24. *A* blacklist *is a pair (B_P, B_N) of sets of set variables $B_P, B_N \subseteq$ SV. We say that a ML pattern φ respects blacklist (B_P, B_N) iff no variable $V_P \in B_P$ occurs positively in φ and no variable $V_N \in B_N$ occurs negatively in φ.*

Lemma 25. *If φ respects blacklist (B_P, B_N), then for any Σ-model $(M, app_M, \{\sigma_M\}_{\sigma \in Sigma})$ and any M-valuation ρ, the function $\mathcal{F}^{M,\rho}_{\varphi,V} \colon \mathcal{P}(M) \to \mathcal{P}(M)$ defined by $\mathcal{F}^{M,\rho}_{\varphi,V}(A) = |\varphi|_{M,\rho[A/V]}$ for any $A \subseteq M$ is antimonotonic for any $V \in B_P$ and monotonic for any $V \in B_N$.*

Proof. Let $A_1 \subseteq A_2 \subseteq M$ We will proceed by structural induction on φ.

- $\varphi \equiv x$:

$$\begin{aligned}\mathcal{F}^{M,\rho}_{\varphi,V}(A_1) &= |x|_{M,\rho[A_1/V]} \\ &= \{\rho(x)\} \\ &= |x|_{M,\rho[A_2/V]} \\ &= \mathcal{F}^{M,\rho}_{\varphi,V}(A_2)\end{aligned}$$

- $\varphi \equiv X$:
  - Let $V \in B_P$. Since X occurs positively in X and X respects (B_N, B_P), it follows that $X \neq V$. Then

$$\begin{aligned}\mathcal{F}^{M,\rho}_{\varphi,V}(A_1) &= |X|_{M,\rho[A_1/V]} \\ &= \rho(X) \\ &= |X|_{M,\rho[A_2/V]} \\ &= \mathcal{F}^{M,\rho}_{\varphi,V}(A_2)\,.\end{aligned}$$

  - Let $V \in B_N$. Then either $X \neq V$, and $\mathcal{F}^{M,\rho}_{\varphi,V}(A_1) = \rho(X) =$

$\mathcal{F}^{M,\rho}_{\varphi,V}(A_2)$ as in the previous case, or $X = V$. Then

$$\begin{aligned}\mathcal{F}^{M,\rho}_{\varphi,V}(A_1) &= |X|_{M,\rho[A_1/V]}\\ &= |X|_{M,\rho[A_1/X]}\\ &= A_1\\ &\subseteq A_2\\ &= |X|_{M,\rho[A_2/X]}\\ &= |X|_{M,\rho[A_2/V]}\\ &= \mathcal{F}^{M,\rho}_{\varphi,V}(A_2)\end{aligned}$$

- $\varphi \equiv \bot$ - trivial.
- $\varphi \equiv \sigma$ - trivial.
- $\varphi \equiv \varphi_1 \to \varphi_2$ - then φ_2 respects (B_P, B_N) and φ_1 respects (B_N, B_P).
  - Let $V \in B_P$. By the induction hypothesis, $\mathcal{F}^{M,\rho}_{\varphi_2,V}$ is antimonotonic and $\mathcal{F}^{M,\rho}_{\varphi_1,V}$ is monotonic. Therefore,

$$\begin{aligned}|\varphi_2|_{M,\rho[A_1/V]} &= \mathcal{F}^{M,\rho}_{\varphi_2,V}(A_1)\\ &\supseteq \mathcal{F}^{M,\rho}_{\varphi_2,V}(A_2)\\ &= |\varphi_2|_{M,\rho[A_2/V]}\end{aligned}$$

    and

$$\begin{aligned}|\varphi_1|_{M,\rho[A_1/V]} &= \mathcal{F}^{M,\rho}_{\varphi_1,V}(A_1)\\ &\subseteq \mathcal{F}^{M,\rho}_{\varphi_1,V}(A_2)\\ &= |\varphi_1|_{M,\rho[A_2/V]}\,.\end{aligned}$$

    Therefore,

$$\begin{aligned}&|\varphi_1|_{M,\rho[A_1/V]} \setminus |\varphi_2|_{M,\rho[A_1/V]}\\ \subseteq\ &|\varphi_1|_{M,\rho[A_2/V]} \setminus |\varphi_2|_{M,\rho[A_2/V]}\end{aligned}$$

    and

$$\begin{aligned}\mathcal{F}^{M,\rho}_{\varphi,V}(A_1) &= |\varphi_1 \to \varphi_2|_{M,\rho[A_1/V]}\\ &= M \setminus (|\varphi_1|_{M,\rho[A_1/V]} \setminus |\varphi_2|_{M,\rho[A_1/V]})\\ &\supseteq M \setminus (|\varphi_1|_{M,\rho[A_2/V]} \setminus |\varphi_2|_{M,\rho[A_2/V]})\\ &= |\varphi_1 \to \varphi_2|_{M,\rho[A_2/V]}\\ &= \mathcal{F}^{M,\rho}\varphi, V(A_2)\,.\end{aligned}$$

  - Let $V \in B_N$. This case is proved similarly.

- $\varphi \equiv \varphi_1\ \varphi_2$ - follows from app_M being a pointwise extension.
- $\varphi \equiv \exists x.\, \varphi'$
  - Let $V \in B_P$. Then

$$\begin{aligned}
\mathcal{F}^{M,\rho}_{\varphi,V}(A_1) &= |\exists x.\, \varphi'|_{M,\rho[A_1/V]} \\
&= \bigcup_{m \in M} |\varphi'|_{M,\rho[A_1/V][m/x]} \\
&= \bigcup_{m \in M} |\varphi'|_{M,\rho[m/x][A_1/V]} \\
&\supseteq \bigcup_{m \in M} |\varphi'|_{M,\rho[m/x][A_2/V]} \\
&= \bigcup_{m \in M} |\varphi'|_{M,\rho[A_2/V][m/x]} \\
&= |\exists x.\, \varphi'|_{M,\rho[A_1/V]} \\
&= \mathcal{F}^{M,\rho}_{\varphi,V}(A_2)
\end{aligned}$$

    where the middle inequality holds because by the induction hypothesis,
    $F^{M,\rho[m/]x}_{\varphi',V}$ is antimonotonic.
  - Let $V \in B_N$. Proved similarly to the previous case.
- $\varphi \equiv \mu X.\, \varphi'$ - Since φ' does not contain a negative occurence of X, φ' respects blacklist $(B_P, B_N \cup \{X\})$.
  - Let $V \in B_P$. We need to prove the inequality in

$$\begin{aligned}
\mathcal{F}^{M,\rho}_{\varphi,V}(A_1) &= |\mu X.\, \varphi'|_{M,\rho[A_1/V]} \\
&= \mu \mathcal{F}^{M,\rho[A_1/V]}_{\varphi',X} \\
&\supseteq \mu \mathcal{F}^{M,\rho[A_2/V]}_{\varphi',X} \\
&= |\mu X.\, \varphi'|_{M,\rho[A_2/V]} \\
&= \mathcal{F}^{M,\rho}_{\varphi,V}(A_2)\,.
\end{aligned}$$

    (The fixpoint exists because by the induction hypothesis, the functions are monotone in X.) By Lemma 23, it is enough to show that for any $B \in M$, $\mathcal{F}^{M,\rho[A_1/V]}_{\varphi',X}(B) \supseteq \mathcal{F}^{M,\rho[A_2/V]}_{\varphi',X}(B)$. If $X \neq V$, then

using the induction hypothesis,

$$\begin{aligned}\mathcal{F}^{M,\rho[A_1/V]}_{\varphi',X}(B) & \\ &= |\varphi'|_{M,\rho[A_1/V][B/X]} \\ &= |\varphi'|_{M,\rho[B/X][A_1/V]} \\ &\supseteq |\varphi'|_{M,\rho[B/X][A_2/V]} \\ &= |\varphi'|_{M,\rho[A_2/V][B/X]} \\ &= \mathcal{F}^{M,\rho[A_1/V]}_{\varphi',X}(B)\,.\end{aligned}$$

On the other hand, if $X = V$, then

$$\mathcal{F}^{M,\rho[A_1/V]}_{\varphi',X}(B) = |\varphi'|_{M,\rho[B/V]} = \mathcal{F}^{M,\rho[A_1/V]}_{\varphi',X}(B)\,.$$

– Let $V \in B_N$. The proof is analogous to the previous case.

□

Lemma 26. *For any ML formula $\mu X.\,\varphi$ and any ML model M, the function $\mathcal{F}^{M,\rho}_{\varphi,X} \colon \mathcal{P}(M) \to \mathcal{P}(M)$ defined as $\mathcal{F}^{M,\rho}_{\varphi,X}(A) = |\varphi|_{M,\rho[A/X]}$ for $A \subseteq M$ is monotone.*

Proof. Since X has no negative occurence in φ, it follows that φ respect blacklist $(\emptyset, \{X\})$ and therefore by Lemma 25, $\mathcal{F}^{M,\rho}_{\varphi,X}$ is monotonic. □

A.2 LTL

The axioms (Prev), (Inf), (NextOut), (NextPFun) and (NextInj) ensure that $\bar{\circ}$ and $\circ$ are interpreted as injective (partial) functions and are inversions of each other. First, they are inversions:

Lemma 27. *For any model M satisfying* (Prev) *and any $m_1, m_2 \in M$,*

$$m_2 \in \mathit{appext}_M(\bar{\circ}_M, \{m_1\}) \iff m_1 \in \mathit{appext}_M(\circ_M, \{m_2\})\,.$$

Proof of Lemma 27. Let $\rho(x) = m_1$. Then $m_2 \in \mathit{appext}_M(\bar{\circ}_M, \{m_1\}) = |\bar{\circ} x|_\rho = |\exists y.\, y \wedge (x \in \circ y)|_\rho$ (by the axiom (Prev)) $= \{m' \in M \mid M = |x \in \circ y|_{\rho[m'/y]}\}$ iff $M = |x \in \circ y|_{\rho[m_2/y]}\}$ iff $m_1 \in \mathit{appext}_M(\circ_M, \{m_2\})$. □

Definition 28. *Let $(M, \mathit{app}_M(\_,\_), \{\sigma_M\}_{\sigma \in \Sigma^{\mathsf{LTL}}})$ be a Γ^{LTL}-model. Let us define:*

- *a partial function $M_\circ : M \rightharpoonup M$ defined by $M_\circ(m) = m'$ whenever m' is the unique element satisfying $\{m'\} = \mathit{appext}_M(\circ_M, \{m\})$; and*
- *a (total) function $M_{\bar{\circ}} : M \to M$ defined by $M_{\bar{\circ}}(m) = m'$, where m' is the unique element satisfying $\{m'\} = \mathit{appext}_M(\bar{\circ}_M, \{m\})$*

for all $m \in M$.

Lemma 29. *Definition 28 is well-formed, and $M_\circ$ and $M_{\bar{\circ}}$ are injective.*

Proof of Lemma 29. By standard matching logic reasoning.

1. $M_\circ$ is a partial function. Let $m \in M_{TrSuf}$ and ρ an M-valuation. Since by the axiom (NextPFun) it holds that

$$\begin{aligned} M &= |\forall x.\, x \in [\![TrSuf]\!] \to \exists y.\, y \in [\![TrSuf]\!] \wedge \circ x \subseteq y|_\rho \\ &= \bigcap_{m' \in M} |x \in [\![TrSuf]\!] \to \exists y.\, y \in [\![TrSuf]\!] \wedge \circ x \subseteq y|_{\rho[m'/x]} \\ &\subseteq |x \in [\![TrSuf]\!] \to \exists y.\, y \in [\![TrSuf]\!] \wedge \circ x \subseteq y|_{\rho[m/x]}\,, \end{aligned}$$

   it follows that

$$\begin{aligned} M &= |\exists y.\, y \in [\![TrSuf]\!] \wedge \circ x \subseteq y|_{\rho[m/x]} \\ &= \bigcup_{m' \in M} |y \in [\![TrSuf]\!] \wedge \circ x \subseteq y|_{\rho[m/x][m'/y]}\,, \end{aligned}$$

   and since $y \in [\![TrSuf]\!] \wedge \circ x \subseteq y$ is a predicate, there must exist some $m' \in M$ such that

$$M = |y \in [\![TrSuf]\!] \wedge \circ x \subseteq y|_{\rho[m/x][m'/y]}\,.$$

   But then $m' \in M_{TrSuf}$ and $appext_M(\circ_M, \{m\}) \subseteq \{m'\}$.

2. $M_\circ$ is injective. Let $m_1, m_2 \in M_{TrSuf}$ such that $M_\circ(m_1) = M_\circ(m_2) = m$ for some $m \in M$. Let ρ be some M-valuation. Then by the axiom (NextInj) it holds that

$$\begin{aligned} M &= |\forall x_1, x_2 \colon TrSuf.\, \circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot \to x_1 = x_2|_\rho \\ &= \bigcap_{m_1' \in M} |x_1 \in [\![TrSuf]\!] \to \forall x_2 \colon TrSuf. \\ &\quad \circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot \to x_1 = x_2|_{\rho[m_1'/x_1]} \\ &\subseteq |x_1 \in [\![TrSuf]\!] \to \forall x_2 \colon TrSuf. \\ &\quad \circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot \to x_1 = x_2|_{\rho[m_1/x_1]}\,. \end{aligned}$$

   Since $m_1 \in [\![TrSuf]\!]$, it follows that

$$\begin{aligned} M = |\forall x_2 \colon TrSuf.\, &\circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot \\ &\to x_1 = x_2|_{\rho[m_1/x_1]}\,. \end{aligned}$$

   By the same argument,

$$M = |\circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot \to x_1 = x_2|_{\rho[m_1/x_1][m_2/x_2]}\,.$$

   Since $M = |\circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot|_{\rho[m_1/x_1][m_2/x_2]}$, it follows that

$$|x_1 = x_2|_{\rho[m_1/x_1][m_2/x_2]}\,;$$

   i.e., $m_1 = m_2$.

3. $M_{\bar\circ}$ is a total function. Let $m \in M_{TrSuf}$. Then by the axiom (Inf) there exists some $m' \in M_{TrSuf}$ such that $m \in \mathit{appext}_M(\circ_M, \{m'\})$. We will show that $\mathit{appext}_M(\bar\circ_M, \{m\}) = \{m'\}$. Since $M_\circ$ is a partial function, it follows that $\{m\} = \mathit{appext}_M(\circ_M, \{m'\})$. By Lemma 27, $m' \in \mathit{appext}_M(\bar\circ_M, \{m\})$. Now we will show that m' is the only member of $\mathit{appext}_M(\bar\circ_M, \{m\})$. Let $m'' \in \mathit{appext}_M(\bar\circ_M, \{m\})$. By Lemma 27, $m \in \mathit{appext}_M(\circ_M, \{m''\})$. Since $m \in [\![\mathit{TrSuf}]\!]$, from the axiom (NextOut) it follows that $m'' \in [\![\mathit{TrSuf}]\!]$, and since $M_\circ$ is a partial function on $[\![\mathit{TrSuf}]\!]$, we have that $m = M_\circ(m'')$. Therefore $M_\circ(M'') = M_\circ(m')$, and from injectivity of $M_\circ(\_)$ it follows that $m'' = m'$, which implies that m' is the only member of $\mathit{appext}_M(\bar\circ_M, \{m\})$.

4. $M_{\bar\circ}(\_)$ is injective. Let $m_1, m_2 \in M_{TrSuf}$ such that $M_{\bar\circ}(m_1) = M_{\bar\circ}(m_2) = m$ for some $m \in M_{TrSuf}$. By Lemma 27 we have $m_1, m_2 \in \mathit{appext}_M(\circ_M, \{m\})$, and since $M_\circ$ is a partial function, it follows that $m_1 = m_2$.

□

Proof of Lemma 7. By structural induction on φ:

- $\varphi \equiv a$ where $a \in \mathrm{AP}$ - $|a|_M \subseteq M_{TrSuf}$ by the axiom (AtomicProp).
- $\varphi \equiv \neg\varphi'$ - $|\mathsf{L2M}(\neg\varphi)|_M = |[\![\mathit{TrSuf}]\!] \wedge \neg\mathsf{L2M}(\varphi)|_M = M_{TrSuf} \cap |\neg\mathsf{L2M}(\varphi)|_M \subseteq [\![\mathit{TrSuf}]\!]$.
- $\varphi \equiv \varphi_1 \wedge \varphi_2$ - $|\mathsf{L2M}(\varphi_1 \wedge \varphi_2)|_M = |\mathsf{L2M}(\varphi_1) \wedge \mathsf{L2M}(\varphi_2)|_M = |\mathsf{L2M}(\varphi_1)|_M \cap |\mathsf{L2M}(\varphi_2)|_M \subseteq M_{TrSuf} \cap M_{TrSuf} = M_{TrSuf}$
- $\varphi \equiv \circ\varphi'$ - $|\mathsf{L2M}(\circ\varphi')|_M = |\circ\mathsf{L2M}(\varphi')|_M = M_\circ(|\mathsf{L2M}(\varphi')|_M) \subseteq M_\circ(M_{TrSuf}) \subseteq M_{TrSuf}$, where the last inclusion is justified as follows. Let $m \in M_\circ(M_{TrSuf})$. Then there exists some $m' \in M_{TrSuf}$ such that $m \in \mathit{appext}_M(\circ_M, \{m'\})$. We will show that $m \in M_{TrSuf}$. Because $M_\circ$ is a partial function (Lemma 29), it follows that $m = M_\circ(m')$. Since $m' \in M_{TrSuf}$, from the axiom (NextPFun) it follows that $m = |\exists y.\, y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y|_{\rho[m'/x]}$ for any M-valuation ρ. Let ρ be such valuation (some valuation exist, because the model M is nonempty). Since $\exists y.\, y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y$ is a predicate, there must exists some $m'' \in M$ such that $|y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y|_{\rho[m'/x][m''/y]} = M$. Therefore, $m'' \in M_{TrSuf}$ and $\circ m' \subseteq \{m''\}$, from which it follows that $\{m\} = \circ m' = \{m''\} \subseteq M_{TrSuf}$, i.e., $m \in M_{TrSuf}$.
- $\varphi \equiv \varphi_1\ U\ \varphi_2$ - We need to show that $|\mathsf{L2M}(\varphi_1\ U\ \varphi_2)|_M = |\mu x.\, \varphi_2 \vee (\varphi_1 \circ X)|_M = \mu F \subseteq M_{TrSuf}$, where $F\colon \mathcal{P}(M) \to \mathcal{P}(M)$ is a function defined by $F(A) = |\varphi_2|_M \cup (|\varphi_1|_M \cap M_\circ(A))$. The inclusion holds because by the Knaster-Tarski theorem, μF is the least prefixpoint of F, and therefore it is contained in M_{TrSuf}, which is a prefixpoint of F: $F(M_{TrSuf}) \subseteq M_{TrSuf}$ holds using the induction hypothesis.

□

Definition 30. *We define a model translation function* $\mathcal{M}^l(\_) : \mathbf{Mod}_{\mathsf{LTL}} \to \mathbf{Mod}_{\mathsf{ML}}(\Gamma^{\mathsf{LTL}})$ *as follows. For an LTL model, i.e., a trace* $\tau \in \mathbf{Mod}_{\mathsf{LTL}}$*, let the carrier set* $\mathcal{M}^l(\tau)$ *be the disjoint union of* $\mathbb{N}_{\geq 1}$ *(the set of positive natural numbers) and* $\{\#\mathsf{def}, \#\mathsf{inh}, \#\mathsf{Trace}, \#\mathsf{TrSuf}, \#\mathsf{next}, \#\mathsf{prev}\}$ *(a set of six distinguished elements). We let* $def_{\mathcal{M}^l(\tau)} = \{\#\mathsf{def}\}$*,* $inh_{\mathcal{M}^l(\tau)} = \{\#\mathsf{inh}\}$*,* $Trace_{\mathcal{M}^l(\tau)} = \{\#\mathsf{Trace}\}$*,* $TrSuf_{\mathcal{M}^l(\tau)} = \{\#\mathsf{TrSuf}\}$*,* $\circ_{\mathcal{M}^l(\tau)} = \{\#\mathsf{next}\}$*,* $\bar{\circ}_{\mathcal{M}^l(\tau)} = \{\#\mathsf{prev}\}$*, and* $a_{\mathcal{M}^l(\tau)} = \{n \in \mathbb{N}_{\geq 1} \mid a \in \tau[n]\}$ *for any* $a \in \mathrm{AP}$*, and define the application as follows:*

1. $app_{\mathcal{M}^l(T)}(\#\mathsf{def}, m) = \mathcal{M}^l(\tau)$ *for any* $m \in \mathcal{M}^h(\tau)$*;*
2. $app_{\mathcal{M}^l(\tau)}(\#\mathsf{inh}, \#\mathsf{Trace}) = \{1\}$*;*
3. $app_{\mathcal{M}^l(\tau)}(\#\mathsf{inh}, \#\mathsf{TrSuf}) = \mathbb{N}_{\geq 1}$*;*
4. $app_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, 1) = \emptyset$*;*
5. $app_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, (n+1)) = \{n\}$ *for all* $n \in \mathbb{N}_{\geq 1}$*;*
6. $app_{\mathcal{M}^l(\tau)}(\#\mathsf{prev}, n) = \{n+1\}$ *for all* $n \in \mathbb{N}_{\geq 1}$*;*
7. *and* $app_{\mathcal{M}^l(\tau)}(a, b) = \emptyset$ *otherwise.*

Proposition 31. *Function* $\mathcal{M}^l(\_)$ *is well-defined, i.e.,* $\mathcal{M}^l(\tau) \models \Gamma^{\mathsf{LTL}}$ *for any LTL model* $\tau \in \mathbf{Mod}_{\mathsf{LTL}}$*.*

Proof of Proposition 31. We will prove the axioms one by one. Let ρ: (EV $\cup$ SV) $\to (\mathcal{M}^l(\tau) \cup \mathcal{P}(\mathcal{M}^l(\tau)))$ be an $\mathcal{M}^l(\tau)$-valuation. Then:

- (Definedness) - $|\lceil x \rceil|_\rho = app_{\mathcal{M}^l(\tau)}(\#\mathsf{def}, |x|_\rho) = app_{\mathcal{M}^l(\tau)}(\#\mathsf{def}, \{\rho(x)\}) = app_{\mathcal{M}^l(\tau)}(\#\mathsf{def}, \rho(x)) = \mathcal{M}^l(\tau)$
- (Prev) - $\mathcal{M}^l(\tau) = |\bar{\circ}x = \exists y.\, y \wedge (x \in \circ y)|_\rho$ iff $|\bar{\circ}x|_\rho = |\exists y.\, y \wedge (x \in \circ y)|_\rho$, which holds because

$$\begin{aligned} |\bar{\circ}x|_\rho &= app_{\mathcal{M}^l(\tau)}(\#\mathsf{prev}, \rho(x)) \\ &= \{\rho(x)+1\} \\ &= \{\rho(x)+1 \mid \rho(x) \in app_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, \rho(x)+1)\} \\ &= \{m+1 \mid \rho(x) \in app_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, m+1)\} \\ &= \{m \mid \rho(x) \in app_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, m)\} \\ &= |\exists y.\, y \wedge (x \in \circ y)|_\rho\,. \end{aligned}$$

  .

- (Trace) - $|\exists x.\, [\![\mathit{Trace}]\!] = x|_\rho \subseteq |[\![\mathit{Trace}]\!] = x|_{\rho[1/x]} = \mathcal{M}^l(\tau)$, because $|\_|_{\rho[1/x]([\![\mathit{Trace}]\!])} = \{1\}$ and $|\_|_{\rho[1/x](x)} = \{1\}$.

- (TraceSuffix) - We want to prove that $|[\![\mathit{TrSuf}]\!] = \mu X.\, [\![\mathit{Trace}]\!] \vee \bar{\circ} X|_\rho = \mathcal{M}^l(\tau)$. Since $|[\![\mathit{TrSuf}]\!]|_\rho = \mathbb{N}_{\geq 1}$, we need to show that $\mathbb{N}_{\geq 1} = \mu F$, where $F(A) = |[\![\mathit{Trace}]\!]|_\rho \cup |\bar{\circ} X|_{\rho[A/X]}$; i.e., $\mathbb{N}_{\geq 1}$ is the least fixpoint of F. First, it is a fixpoint:

$$\begin{aligned} F(\mathbb{N}_{\geq 1}) &= \{1\} \cup |\exists y.\, X \in \circ y|_{\rho[\mathbb{N}_{\geq 1}/X]} \\ &= \{1\} \cup \{m \in \mathcal{M}^l(\tau) \mid \\ &\quad |X \in \circ y|_{\rho[\mathbb{N}_{\geq 1}/X][m/y]} = \mathcal{M}^l(\tau)\} \\ &= \{1\} \cup \{m \in \mathcal{M}^l(\tau) \mid \exists m' \in \mathbb{N}_{\geq 1}. \\ &\quad m' \in \mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, m)\} \\ &= \{1\} \cup \{m + 1 \mid m \in \mathbb{N}_{\geq 1}\} \\ &= \mathbb{N}_{\geq 1}\,. \end{aligned}$$

  It is the least fixpoint. Let $A = F(A)$. We want to show that $\mathbb{N}_{\geq 1} \subseteq A$. We will proceed by induction.

  - $\{1\} = \mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{inh}, \#\mathsf{Trace}) = |[\![\mathit{Trace}]\!]|_\rho \subseteq F(A) = A$.
  - Assuming $n \in \mathbb{N}_{\geq 1}$, we will show that $n + 1 \in A = F(A)$. It holds if $n{+}1 \in |\bar{\circ} X|_{\rho[A/X]}$, iff $n{+}1 \in \bar{\circ}_{\mathcal{M}^l(\tau)}(A)$ iff $\exists m \in A.\, n{+}1 \in \bar{\circ}_{\mathcal{M}^l(\tau)}(m)$ iff $\exists m \in A.\, m \in \mathcal{M}^l(\tau)_\circ(n+1)$, which holds by the construction and choice $m = n$.

- (NextOut) - $\mathcal{M}^l(\tau) = |\circ(\neg[\![\mathit{TrSuf}]\!]) \subseteq \neg[\![\mathit{TrSuf}]\!]|_\rho$ iff $|\circ(\neg[\![\mathit{TrSuf}]\!])|_\rho \subseteq |\neg[\![\mathit{TrSuf}]\!]|_\rho$. But $|\circ(\neg[\![\mathit{TrSuf}]\!])|_\rho = \mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, |\neg[\![\mathit{TrSuf}]\!]|_\rho) = \emptyset$, since for any $m \in |\neg[\![\mathit{TrSuf}]\!]|_\rho = \mathcal{M}^l(\tau) \setminus \mathbb{N}_{\geq 1}$, $\mathit{app}_M(\#\mathsf{next}, m) = \emptyset$.

- (Inf) - $|[\![\mathit{TrSuf}]\!] \subseteq \circ[\![\mathit{TrSuf}]\!]|_\rho = \mathcal{M}^l(\tau)$, because $|[\![\mathit{TrSuf}]\!]|_\rho = \mathbb{N}_{\geq 1} \subseteq \mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, \mathbb{N}_{\geq 1}) = |\circ[\![\mathit{TrSuf}]\!]|_\rho$: for any $x \in \mathbb{N}_{\geq 1}$, we have $\{x\} = \mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, (x + 1))$.

- (NextPFun) - We have

$$\begin{aligned} &|\circ\colon \mathit{TrSuf} \rightharpoonup \mathit{TrSuf}|_\rho \\ &= |\forall x.\, x \in [\![\mathit{TrSuf}]\!] \to \exists y.\, y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y|_\rho \\ &= \bigcap_{m \in \mathcal{M}^l(\tau)} |x \in [\![\mathit{TrSuf}]\!] \to \exists y.\, y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y|_{\rho[m/x]} \\ &= \mathcal{M}^l(\tau)\,, \end{aligned}$$

  because for any $m \in \mathcal{M}^l(\tau)$, we have

$$|x \in [\![\mathit{TrSuf}]\!] \to \exists y.\, y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y|_{\rho[m/x]} = \mathcal{M}^l(\tau)\,,$$

because for any $n \in \mathbb{N}_{\geq 1}$,

$$\begin{aligned}
&|\exists y.\, y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y|_{\rho[n/x]} \\
&= \bigcup_{m' \in \mathcal{M}^l(\tau)} |y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y|_{\rho[n/x][m'/y]} \\
&= \mathcal{M}^l(\tau)\,.
\end{aligned}$$

Indeed, choose $n \in \mathbb{N}_{\geq 1}$ arbitrarily. If $n = 1$, then

$$\begin{aligned}
&\bigcup_{m' \in \mathcal{M}^l(\tau)} |y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y|_{\rho[n/x][m'/y]} \\
&\supseteq |y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y|_{\rho[1/x][1/y]}\,,
\end{aligned}$$

because $1 \in \mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{inh}, \mathit{TrSuf})$ and $\mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, 1) = \emptyset \subseteq \{1\}$. If $n = n' + 1$ for some $n \in \mathbb{N}_{\geq 1}$, we have

$$\begin{aligned}
&\bigcup_{m' \in \mathcal{M}^l(\tau)} |y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y|_{\rho[n/x][m'/y]} \\
&\supseteq |y \in [\![\mathit{TrSuf}]\!] \wedge \circ x \subseteq y|_{\rho[(n'+1)/x][n/y]}\,,
\end{aligned}$$

because $n' \in \mathbb{N}_{\geq 1} = \mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{inh}, \mathit{TrSuf})$, and $\mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, (n'+1)) = \{n'\} \subseteq \{n'\}$.

- (NextInj) - $|\forall x_1, x_2\colon \mathit{TrSuf}.\, \circ x_1 = \circ x_2 \wedge x_1 \neq \bot \rightarrow x_1 = x_2|_\rho = \bigcap_{m_1 \in \mathcal{M}^l(\tau)} |\text{-}|_{\rho[m_1/x_1]}(x_1 \in [\![\mathit{TrSuf}]\!] \rightarrow \forall x_2\colon \mathit{TrSuf}.\, \circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot \rightarrow x_1 = x_2) = \mathcal{M}^l(\tau)$, because for any $m \in \mathcal{M}^l(\tau)$, $|\text{-}|_{\rho[m_1/x_1]}(x_1 \in [\![\mathit{TrSuf}]\!] \rightarrow \forall x_2\colon \mathit{TrSuf}.\, \circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot \rightarrow x_1 = x_2) = \mathcal{M}^l(\tau)$. Choose m_1 arbitrarily. Then either $m_1 \notin \mathbb{N}_{\geq 1}$, and the equality trivially holds, or $m_1 \in \mathbb{N}_{\geq 1}$, in which case we need to show that

  $$|\text{-}|_{\rho[m_1/x_1]}(\forall x_2\colon \mathit{TrSuf}.\, \circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot \rightarrow x_1 = x_2) = \mathcal{M}^l(\tau)\,.$$

  By similar argument, we can choose $m_2 \in \mathbb{N}_{\geq 1}$ arbitrarily and prove that $|\text{-}|_{\rho[m_1/x_1][m_2/x_2]}(\circ x_1 = \circ x_2 \wedge \circ x_1 \neq \bot \rightarrow x_1 = x_2) = \mathcal{M}^l(\tau)$. We need to show that $m_1 = m_2$, assuming that $\mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, m_1) \neq \emptyset$ and that $\mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, m_1) = \mathit{app}_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, m_2)$. But from the definition of $\mathit{app}_{\mathcal{M}^l(\tau)}(\text{-}, \text{-})$ it follows that $m_1 = n_1 + 1$, $m_2 = n_2 + 1$ and $n_1 = n_2$ for some $n_1, n_2 \in \mathbb{N}_{\geq 1}$. But then $m_1 = m_2$.

- (AtomicProp) - $|a \subseteq [\![\mathit{TrSuf}]\!]|_\rho = \mathcal{M}^l(\tau)$ by construction.

□

Proof of Lemma 8. We will proceed by structural induction on φ.

- $\varphi \equiv a \text{ - } \tau, i \vDash_{\mathsf{LTL}} a$ iff $\vDash_{\mathsf{LTL}}, a \in \tau[i]$ (by the definition of $\vDash_{\mathsf{LTL}}$) iff $i \in \mathcal{M}^l(\tau)_a$ (by the construction of $\mathcal{M}^l(\tau)$) iff $i \in |a|_{\mathcal{M}^l(\tau)}$ (by the definition of $\vDash$).

- $\varphi \equiv \neg\varphi'$ - $\tau, i \vDash_{\mathsf{LTL}} \neg\varphi'$ iff $\tau, i \nvDash_{\mathsf{LTL}} \varphi'$ (by the definition of $\vDash_{\mathsf{LTL}}$) iff $i \notin |\mathsf{L2M}(\varphi')|_{\mathcal{M}^l(\tau)}$ (by the induction hypothesis) iff $i \in |\neg\mathsf{L2M}(\varphi')|_{\mathcal{M}^l(\tau)}$ (by the definition of $\vDash$) iff $i \in |\neg\mathsf{L2M}(\varphi')|_{\mathcal{M}^l(\tau)} \cap \mathcal{M}^l(\tau)_{TrSuf}$ (because $i \in \mathcal{M}^l(\tau)_{TrSuf}$ by the construction of $\mathcal{M}^l(\tau)$) iff $i \in |\neg\mathsf{L2M}(\varphi') \wedge [\![TrSuf]\!]|_{\mathcal{M}^l(\tau)}$ iff $i \in |\mathsf{L2M}(\neg\varphi')|_{\mathcal{M}^l(\tau)}$ (by desugaring notations).

- $\varphi \equiv \varphi_1 \wedge \varphi_2$ - $\tau, i \vDash_{\mathsf{LTL}} \varphi_1 \wedge \varphi_2$ iff $\tau, i \vDash_{\mathsf{LTL}} \varphi_1$ and $\tau, i \vDash_{\mathsf{LTL}} \varphi_2$ (by the definition of $\vDash_{\mathsf{LTL}}$) iff $i \in |\mathsf{L2M}(\varphi_1)|_{\mathcal{M}^l(\tau)}$ and $i \in |\mathsf{L2M}(\varphi_2)|_{\mathcal{M}^l(\tau)}$ (by the induction hypothesis) iff $i \in |\mathsf{L2M}(\varphi_1)|_{\mathcal{M}^l(\tau)} \cup |\mathsf{L2M}(\varphi_2)|_{\mathcal{M}^l(\tau)}$ iff $i \in |\mathsf{L2M}(\varphi_1) \wedge \mathsf{L2M}(\varphi_2)|_{\mathcal{M}^l(\tau)}$ iff $i \in |\mathsf{L2M}(\varphi_1 \wedge \varphi_2)|_{\mathcal{M}^l(\tau)}$.

- $\varphi \equiv \circ\varphi'$ - $\tau, i \vDash_{\mathsf{LTL}} \circ\varphi'$ iff $\tau, i{+}1 \vDash_{\mathsf{LTL}} \varphi'$ (by the definition of $\vDash_{\mathsf{LTL}}$) iff $i{+}1 \in |\mathsf{L2M}(\varphi')|_{\mathcal{M}^l(\tau)}$ (by the induction hypothesis) iff $i \in app_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, i+1) \wedge i{+}1 \in |\mathsf{L2M}(\varphi')|_{\mathcal{M}^l(\tau)}$ (by the definition of $\mathcal{M}^l(\tau)$) iff $\exists j \in |\mathsf{L2M}(\varphi')|_{\mathcal{M}^l(\tau)}. i \in app_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, j)$ (by the definition of $\mathcal{M}^l(\tau)$) iff $i \in \bigcup\{ app_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, j) \mid j \in |\mathsf{L2M}(\varphi')|_{\mathcal{M}^l(\tau)}\}$ iff $i \in app_{\mathcal{M}^l(\tau)}(\#\mathsf{next}, |\mathsf{L2M}(\varphi')|_{\mathcal{M}^l(\tau)})$ iff $i \in |\circ\mathsf{L2M}(\varphi')|_{\mathcal{M}^l(\tau)}$ iff $i \in |\mathsf{L2M}(\circ\varphi')|_{\mathcal{M}^l(\tau)}$.

- $\varphi \equiv \varphi_1 \, U \, \varphi_2$ - Since $\varphi_1 \, U \, \varphi_2$ is a notation for $\mu X.\varphi_2 \vee (\varphi_1 \wedge \circ X)$, in ML we have that $|\mathsf{L2M}(\varphi_1 \, U \, \varphi_2)|_{\mathcal{M}^l(\tau)} = \mu F_U$, where $F_U(\_) : \mathcal{P}(\mathcal{M}^l(\tau)) \to \mathcal{P}(\mathcal{M}^l(\tau))$ is defined by $F_U(A) = |\mathsf{L2M}(\varphi_2)|_{\mathcal{M}^l(\tau)} \cup (|\mathsf{L2M}(\varphi_1)|_{\mathcal{M}^l(\tau)} \cap (appext_{\mathcal{M}^l(\tau)}(\{\#\mathsf{next}\}, A)))$. Because $\mu F_U = F_U(\mu F_U)$, i.e. μF_U is a fixpoint of F_U, we can expand

$$\begin{aligned} &|\mathsf{L2M}(\varphi_1 \, U \, \varphi_2)|_{\mathcal{M}^l(\tau)} \\ &= |\mathsf{L2M}(\varphi_2)|_{\mathcal{M}^l(\tau)} \cup (|\mathsf{L2M}(\varphi_1)|_{\mathcal{M}^l(\tau)} \\ &\cap (appext_{\mathcal{M}^l(\tau)}(\{\#\mathsf{next}\}, |\mathsf{L2M}(\varphi_1 \, U \, \varphi_2)|_{\mathcal{M}^l(\tau)})) \, . \end{aligned} \tag{1}$$

For $\tau, i \vDash_{\mathsf{LTL}} \varphi_1 \, U \, \varphi_2$ implies $i \in |\mathsf{L2M}(\varphi_1 \, U \, \varphi_2)|_{\mathcal{M}^l(\tau)}$, we first prove a stronger statement:

Claim 32. *For all $m, n \in \mathbb{N}$ such that $n \leq m$,*

$$\begin{aligned} &(\tau, m \vDash_{\mathsf{LTL}} \varphi_2 \wedge \forall o < n. \, \tau, m - o - 1 \vDash_{\mathsf{LTL}} \varphi_1) \\ &\implies \forall p \leq n. \, (m - p) \in |\mathsf{L2M}(\varphi_1 \, U \, \varphi_2)|_{\mathcal{M}^l(\tau)} \, . \end{aligned}$$

Proof. By induction on n.

- $n = 0$ - Assuming $\tau, m \vDash_{\mathsf{LTL}} \varphi_2$, by the (outer) induction hypothesis $m \in |\mathsf{L2M}(\varphi_2)|_{\mathcal{M}^l(\tau)}$, and from (1) it follows that $(m{-}0) \in |\mathsf{L2M}(\varphi_1 U \varphi_2)|_{\mathcal{M}^l(\tau)}$.
- $n > 0$ - Assuming the induction hypothesis

$$\begin{aligned} &(\tau, m \vDash_{\mathsf{LTL}} \varphi_2 \wedge \forall o < n - 1. \, \tau, m - o - 1 \vDash_{\mathsf{LTL}} \varphi_1) \\ &\implies \forall p \leq n - 1. \, (m - p) \in |\mathsf{L2M}(\varphi_1 \, U \, \varphi_2)|_{\mathcal{M}^l(\tau)} \end{aligned}$$

and assuming $\tau, m \vDash_{\mathsf{hLTL}} \varphi_2$ and $\forall o < n.\, \tau, m-o-1 \vDash_{\mathsf{LTL}} \varphi_2$ it follows that $\forall p \leq n-1.\, (m-p) \in |\mathsf{L2M}(\varphi_1\, U\, \varphi_2)|_{\mathcal{M}^\iota(\tau)}$. It remains to prove the case when $p = n$, i.e. $(m-n) \in |\mathsf{L2M}(\varphi_1\, U\, \varphi_2)|_{\mathcal{M}^\iota(\tau)}$. Because of the equation (1) it is enough to prove that $(m-n) \in |\mathsf{L2M}(\varphi_1)|_{\mathcal{M}^\iota(\tau)}$ and $(m-n) \in \mathit{appext}_{\mathcal{M}^\iota(\tau)}(\{\#\mathsf{next}\}, |\mathsf{L2M}(\varphi_1\, U\, \varphi_2)|_{\mathcal{M}^\iota(\tau)})$. For the former, we use the outer induction hypothesis and the assumption $\tau, m-(n-1)-1 \vDash_{\mathsf{LTL}} \varphi_1$, while the latter holds because by the definition of $\mathit{appext}_{\mathcal{M}^\iota(\tau)}(\_, \_)$ and because by the (inner) induction hypothesis, $(m-(n-1)) \in |\mathsf{L2M}(\varphi_1\, U\, \varphi_2)|_{\mathcal{M}^\iota(\tau)}$.

□

Now assume $\tau, i \vDash_{\mathsf{LTL}} \varphi_1\, U\, \varphi_2$. By the definition of $\vDash_{\mathsf{LTL}}$, there exists $j \geq i$ such that $\tau, j \vDash_{\mathsf{hLTL}} \varphi_2$ and for all $i \leq k < j$ we have $\tau, k \vDash_{\mathsf{hLTL}} \varphi_1$. When in the above claim we choose m to be j, n to be $j-i$ and p to be n, we get $i = j-(j-i) \in |\mathsf{L2M}(\varphi_1\, U\, \varphi_2)|_{\mathcal{M}^\iota(\tau)}$.

For the other implication, that $\tau, i \nvDash_{\mathsf{LTL}} \varphi_1\, U\, \varphi_2$ implies $i \notin |\mathsf{L2M}(\varphi_1\, U\, \varphi_2)|_{\mathcal{M}^\iota(\tau)}$, we first prove a stronger statement:

Claim 33. *For all $m, n \in \mathbb{N}$ such that $n \leq m$,*

$$\begin{aligned} &(\tau, m \nvDash_{\mathsf{LTL}} \varphi_1 \wedge \forall o \leq n.\, \tau, m-o \nvDash_{\mathsf{LTL}} \varphi_2) \\ \implies\ &\forall p \leq n.\, (m-p) \notin |\mathsf{L2M}(\varphi_1\, U\, \varphi_2)|_{\mathcal{M}^\iota(\tau)}\,. \end{aligned}$$

Proof. By induction on n.

- $n = 0$ - Assuming $\tau, m \nvDash_{\mathsf{LTL}} \varphi_1$ and $\tau, m-0 \nvDash_{\mathsf{LTL}} \varphi_1$, by the (outer) induction hypothesis it holds that $m \notin |\mathsf{L2M}(\varphi_1)|_{\mathcal{M}^\iota(\tau)}$ and $m \notin |\varphi_2|_{\mathcal{M}^\iota(\tau)}$. But then by (1), $m \notin |\mathsf{L2M}(\psi_1\, U\, \psi_2)|_{\mathcal{M}^\iota(\tau)}$.
- $n > 0$ - Assume $\tau, m \nvDash_{\mathsf{LTL}} \varphi_1$ and $\forall o \leq n.\, \tau, m-o \nvDash_{\mathsf{hLTL}} \varphi_2$. From the inner induction hypothesis it follows that $\forall p \leq n-1.\, (m-p) \notin |\mathsf{L2M}(\varphi_1\, U\, \varphi_2)|_{\mathcal{M}^\iota(\tau)}$. It remains to prove the case where $p = n$, that $(m-n) \notin |\mathsf{L2M}(\varphi_1\, U\, \varphi_2)|_{\mathcal{M}^\iota(\tau)}$. Since $\tau, m-n \nvDash_{\mathsf{hLTL}} \psi_2$, from the outer induction hypothesis it follows that $(m-n) \notin |\mathsf{L2M}(\varphi_2)|_{\mathcal{M}^\iota(\tau)}$. By (1), it is now enough to prove that $(m-n) \notin \mathit{appext}_{\mathcal{M}^\iota(\tau)}(\{\#\mathsf{next}\}, |\mathsf{L2M}(\varphi_1\, U\, \varphi_2))|_{\mathcal{M}^\iota(\tau)})$, which is true if $(m-(n-1)) \notin |\mathsf{L2M}(\psi_1\, U\, \psi_2)|_{\mathcal{M}^\iota(\tau)}$, which holds by the (inner) induction hypothesis.

□

Now assume $\tau, i \nvDash_{\mathsf{LTL}} \varphi_1\, U\, \varphi_2$. By definition of $\vDash_{\mathsf{LTL}}$, either there is no $j \geq i$ satisfying $\tau, j \vDash_{\mathsf{hLTL}} \psi_2$, in which case $\emptyset$ is a fixpoint of F_U, or strictly before first such j there exists some k, $i \leq k < j$ satisfying $\tau, k \nvDash_{\mathsf{LTL}} \varphi_1$ (and by choice of j also $\tau, k \nvDash_{\mathsf{LTL}} \varphi_2$). Then we can use k and $k-i$ as parameters m and n of the claim above and by the choice $p = k-i$ get $i = k-(k-i) \notin |\mathsf{L2M}(\varphi_1\, U\, \varphi_2)|_{\mathcal{M}^\iota(\tau)}$.

□

Proof of Theorem 9.

$$\begin{aligned}
\tau \vDash_{\mathsf{LTL}} \varphi &\iff \tau, 1 \vDash_{\mathsf{LTL}} \varphi \\
&\iff 1 \in |\mathsf{L2M}(\varphi)|_{\mathcal{M}^l(\tau)} \text{ (by Lemma 8)} \\
&\iff \mathcal{M}^l(\tau)_{\mathit{Trace}} \in |\mathsf{L2M}(\varphi)|_{\mathcal{M}^l(\tau)} \\
&\quad \text{(since } 1 = \mathcal{M}^l(\tau)_{\mathit{Trace}} \text{ by construction)} \\
&\iff \mathcal{M}^l(\tau) \vDash [\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi)\,.
\end{aligned}$$

□

Lemma 34. *Let M be a model of Γ^{LTL}, and $m \in M_{\mathit{TrSuf}}$. A sequence $m_1, m_2, \ldots m_n = m$ where $m_i \in M$ such that $m_1 = M_{\mathit{Trace}}$ and $m_{i+1} \in M_{\bar{\circ}}(m_i)$ for all $1 \leq i < n$ is called an initial sequence of m. For every m there exists exactly one such sequence, and we define a function $\mathit{dist}_M^{\mathsf{LTL}} \colon M_{\mathit{TrSuf}} \to \mathbb{N}_{\geq 1}$ by $\mathit{dist}_M^{\mathsf{LTL}}(m) = n$.*

Proof of Lemma 34. For existence, the axiom (TraceSuffix) enforces that $M_{\mathit{TrSuf}} = \mu F$, where $F(A) = M_{\mathit{Trace}} \cup M_{\bar{\circ}}(A)$. Define:

$$\begin{aligned}
\xi = \{m \mid \exists n \in \mathbb{N}_{\geq 1}, \exists m_1, \ldots, m_n.\, & m_1 = M_{\mathit{Trace}} \\
& \wedge m_n = m \wedge \forall 1 \leq i < n.\, m_{i+1} = M_{\bar{\circ}}(m_i)\}
\end{aligned}$$

and since $F(\xi) \subseteq \xi$, i.e. ξ is a prefix point of F, from the Knaster-Tarski theorem it follows that $M \subseteq \xi$, i.e. for every $m \in M$ there exists an appropriate sequence.

For uniqueness, let there be two such sequences, $m_1, \ldots, m_n$ and $m'_1, \ldots, m'_{n'}$, and let l be the length of their maximal common suffix, starting with $m_{n-(l-1)} = m'_{n'-(l-1)}$. Let us assume (w.l.o.g.) that $n \geq n'$. It must be true that $l = n'$, because if $l < n'$, then by the injectivity of $M_{\bar{\circ}}$ (Lemma 38) there is another common suffix starting with $m_{n-l} = m'_{m'-l}$, which contradicts our choice of maximal common suffix. Since $l = n'$, we have that $m_{n-(l-1)} = m'_{n'-(l-1)} = m'_1 \in M_{\mathit{Trace}}$ But then $M_{\circ}(m_{n-(l-1)})$ is undefined, therefore $m_{n-(l-1)}$ has no predecessors in the sequence $m_1, \ldots, m_n$, and $l = n = n'$, and the two sequences are identical. □

Proof of Lemma 12. We want to prove that

$$\forall n \in \mathbb{N}_{\geq 1}.\, \mathit{dist}_M^{\mathsf{LTL}}([\![n]\!]_M^{\mathsf{LTL}}) = n \tag{2}$$

and that

$$\forall m \in M_{\mathit{TrSuf}}.\, [\![\mathit{dist}_M^{\mathsf{LTL}}(m)]\!]_M^{\mathsf{LTL}} = m\,. \tag{3}$$

First, we will prove (2) by induction on n. For the case when $n = 1$, we have $[\![1]\!]_M^{\mathsf{LTL}} \subseteq [\![\mathit{Trace}]\!]$, therefore $[\![1]\!]_M^{\mathsf{LTL}}$ is the (unique) initial sequence of $[\![1]\!]_M^{\mathsf{LTL}}$, and therefore $\mathit{dist}_M^{\mathsf{LTL}}([\![1]\!]_M^{\mathsf{LTL}}) = 1$. For the case when $n = n' + 1$, we assume the induction hypothesis $\mathit{dist}_M^{\mathsf{LTL}}([\![n']\!]_M^{\mathsf{LTL}}) = n'$, from which it follows

that $dist_M^{\mathsf{LTL}}(ts_M(n')) = \{n'\}$. By the definition of $[\![\_]\!]_M^{\mathsf{LTL}}$, we have $\{n'+1\} = M_{\bar{\circ}}(ts_M(n'))$; this element extends the initial sequence of $[\![n']\!]_M^{\mathsf{LTL}}$ into the initial sequence of $[\![n'+1]\!]_M^{\mathsf{LTL}}$. Therefore, $dist_M^{\mathsf{LTL}}([\![n'+1]\!]_M^{\mathsf{LTL}}) = dist_M^{\mathsf{LTL}}([\![n']\!]_M^{\mathsf{LTL}}) + 1 = n'+1$. . Second, we will prove (3) by induction on $dist_M^{\mathsf{LTL}}(m)$. For the case when $dist_M^{\mathsf{LTL}}(m) = 1$, by the definition in Lemma 34, we have $m = M_{Trace}$, we have $[\![1]\!]_M^{\mathsf{LTL}} = M_{Trace}$ by definition. For the case when $dist_M^{\mathsf{LTL}}(m) = n+1$, by the definition of $dist_M^{\mathsf{LTL}}$, there exists some $m' \in M_{TrSuf}$ such that $m \in M_{\bar{\circ}}(m')$ and $dist_M^{\mathsf{LTL}}(m') = n$. By the induction hypothesis, $[\![n]\!]_M^{\mathsf{LTL}} = [\![dist_M^{\mathsf{LTL}}(m')]\!]_M^{\mathsf{LTL}} = m'$. Therefore, $[\![dist_M^{\mathsf{LTL}}(m)]\!]_M^{\mathsf{LTL}} = [\![n+1]\!]_M^{\mathsf{LTL}} = m''$ where $\{m''\} = ts_M(n+1) = M_{\bar{\circ}}(ts_M(n)) = M_{\bar{\circ}}(\{m'\}) = M_{\bar{\circ}}(m')$. So we have $m, m'' \in M_{\bar{\circ}}(m')$, and since $M_{\bar{\circ}}(\_)$ returns a singleton set (Lemma 29), it follows that $m = m''$ and therefore $[\![dist_M^{\mathsf{LTL}}(m)]\!]_M^{\mathsf{LTL}} = m$.

□

Proof of Lemma 13. By structural induction on φ.

- For $\varphi \equiv a$,

$$\begin{aligned}
|\mathsf{L2M}(a)|_{\mathcal{M}^l(\mathcal{T}^l(M))} &= |a|_{\mathcal{M}^l(\mathcal{T}^l(M))} \\
&= \{n \in \mathbb{N}_{\geq 1} \mid a \in \mathcal{T}^l(M)[n]\} \\
&= \{n \in \mathbb{N}_{\geq 1} \mid [\![n]\!]_M^{\mathsf{LTL}} \in M_a\} \\
&= \{dist_M^{\mathsf{LTL}}(m) \mid m \in M_a\} \\
&= dist_M^{\mathsf{LTL}}(M_a) \\
&= dist_M^{\mathsf{LTL}}(|a|_M) \\
&= dist_M^{\mathsf{LTL}}(|\mathsf{L2M}(a)|_M)\,,
\end{aligned}$$

  where the fourth equality holds for the following reason: $n \in \{n \in \mathbb{N}_{\geq 1} \mid [\![n]\!]_M^{\mathsf{LTL}} \in M_a\}$ iff $[\![n]\!]_M^{\mathsf{LTL}} \in M_a$ iff $n = dist_M^{\mathsf{LTL}}([\![n]\!]_M^{\mathsf{LTL}}) \in \{dist^{\mathsf{LTL}}(m) \mid m \in M_a\}$, where $n = dist_M^{\mathsf{LTL}}([\![n]\!]_M^{\mathsf{LTL}})$ by Lemma 12.

- For $\varphi \equiv \neg\varphi'$,

$$\begin{aligned}
&|\mathsf{L2M}(\neg\varphi')|_{\mathcal{M}^l(\mathcal{T}^l(M))} \\
&= |\neg\mathsf{L2M}(\varphi') \wedge [\![\mathit{TrSuf}]\!]|_{\mathcal{M}^l(\mathcal{T}^l(M))} \\
&= \mathcal{M}^l(\mathcal{T}^l(M))_{\mathit{TrSuf}} \setminus |\mathsf{L2M}(\varphi')|_{\mathcal{M}^l(\mathcal{T}^l(M))} \\
&= \mathbb{N}_{\geq 1} \setminus dist_M^{\mathsf{LTL}}|\mathsf{L2M}(\varphi')|_M \\
&= dist_M^{\mathsf{LTL}}([\![\mathbb{N}_{\geq 1}]\!]_M^{\mathsf{LTL}}) \setminus dist_M^{\mathsf{LTL}}(|\mathsf{L2M}(\varphi')|_M) \\
&= dist_M^{\mathsf{LTL}}([\![\mathbb{N}_{\geq 1}]\!]_M^{\mathsf{LTL}} \setminus |\mathsf{L2M}(\varphi')|_M) \\
&= dist_M^{\mathsf{LTL}}(M_{\mathit{TrSuf}} \setminus |\mathsf{L2M}(\varphi')|_M) \\
&= idx_M^{\mathsf{LTL}}(\{m \in M_{\mathit{TrSuf}} \mid m \notin |\mathsf{L2M}(\varphi')|_M\}) \\
&= dist_M^{\mathsf{LTL}}(|\mathsf{L2M}(\varphi') \wedge [\![\mathit{TrSuf}]\!]|_M) \\
&= dist_M^{\mathsf{LTL}}(|\mathsf{L2M}(\neg\varphi')|_M)\,.
\end{aligned}$$

- For $\varphi \equiv \varphi_1 \wedge \varphi_2$, the result follows by simplifications and using the induction hypothesis.
- For $\varphi \equiv \circ\varphi'$,

$$
\begin{aligned}
&|\mathsf{L2M}(\circ\varphi')|_{\mathcal{M}^l(\mathcal{T}^l(M))} \\
&= |\circ\mathsf{L2M}(\varphi')|_{\mathcal{M}^l(\mathcal{T}^l(M))} \\
&= \mathit{appext}_{\mathcal{M}^l(\mathcal{T}^l(M))}(\{\#\mathsf{next}\}, |\mathsf{L2M}(\varphi')|^l_{\mathcal{M}}(\mathcal{T}^l(M))) \\
&= \mathit{app}_{\mathcal{M}^l(\mathcal{T}^l(M))}(\#\mathsf{next}, \\
&\mathit{dist}^{\mathsf{LTL}}_M(|[\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi')|_M)) \\
&= \mathit{dist}^{\mathsf{LTL}}_M(M_\circ(|\mathsf{L2M}(\varphi')|_M)) \\
&= \mathit{dist}^{\mathsf{LTL}}_M(\mathit{appext}_M(|\circ|_M, |\mathsf{L2M}(\varphi')|_M)) \\
&= \mathit{dist}^{\mathsf{LTL}}_M(|\circ\mathsf{L2M}(\varphi')|_M) \\
&= \mathit{dist}^{\mathsf{LTL}}_M(|\mathsf{L2M}(\circ\varphi')|_M)\,,
\end{aligned}
$$

  where the fourth equality holds for the following reason:

$$
\begin{aligned}
&n \in \mathit{app}_{\mathcal{M}^l(\mathcal{T}^l(M))}(\#\mathsf{next}, \\
&\mathit{dist}^{\mathsf{LTL}}_M(|[\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi')|_M)) \\
&\quad\iff n+1 \in \mathit{dist}^{\mathsf{LTL}}_M(|[\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi')|_M) \\
&\quad\iff \exists m \in |[\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi')|_M. \\
&\qquad\quad n+1 = \mathit{dist}^{\mathsf{LTL}}_M(m) \\
&\quad\iff \exists m \in |[\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi')|_M. \\
&\qquad\quad \exists m' \in M_\circ(m).\, n+1 = \mathit{dist}^{\mathsf{LTL}}_M(m)) \\
&\quad\iff \exists m \in |[\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi')|_M. \\
&\qquad\quad \exists m' \in M_\circ(m).\, n = \mathit{dist}^{\mathsf{LTL}}_M(m')) \\
&\quad\iff \exists m' \in M_\circ(|[\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi')|_M). \\
&\qquad\quad n = \mathit{dist}^{\mathsf{LTL}}_M(m') \\
&\quad\iff n \in \mathit{dist}^{\mathsf{LTL}}_M(M_\circ(|\mathsf{L2M}(\varphi')|_M)).
\end{aligned}
$$

- For $\varphi \equiv \varphi_1\, U\, \varphi_2$, by definition of the U operator, we have

$$
\begin{aligned}
&\mathit{dist}^{\mathsf{LTL}}_M(|\mathsf{L2M}(\varphi_1\, U\, \varphi_2)|_M) \\
&= \mathit{dist}^{\mathsf{LTL}}_M(|\mu X\,.\varphi_2 \vee (\varphi_1 \wedge \circ X)|_M) \\
&= \mathit{dist}^{\mathsf{LTL}}_M(\mu F)\,,
\end{aligned}
$$

  where

$$F(X) = |\varphi_2|_M \cup (|\varphi_1|_M \cap M_\circ(X))\,.$$

We also have

$$\begin{aligned}&|\mathsf{L2M}(\varphi_1\ U\ \varphi_2)|_{\mathcal{M}^l(\mathcal{T}^l(M))}\\&= |\mu X\,.\varphi_2 \vee (\varphi_1 \wedge \circ X)|_{\mathcal{M}^l(\mathcal{T}^l(M))}\\&= \mu G\,,\end{aligned}$$

where

$$\begin{aligned}G(X) &= |\varphi_2|_{\mathcal{M}^l(\mathcal{T}^l(M))}\\&\quad \cup (|\varphi_1|_{\mathcal{M}^l(\mathcal{T}^l(M))} \cap \mathcal{M}^l(\mathcal{T}^l(M))_\circ(X))\\&\quad = \mathit{dist}_M^{\mathsf{LTL}}(|\varphi_2|_M) \cup\\&\quad (\mathit{dist}_M^{\mathsf{LTL}}(|\varphi_1|_M) \cap \mathit{dist}_M^{\mathsf{LTL}}(M_\circ([\![X]\!]_M)))\\&\quad = \mathit{dist}_M^{\mathsf{LTL}}(|\varphi_2|_M \cup (|\varphi_1|_M \cap M_\circ([\![X]\!]_M)))\,,\end{aligned}$$

where the second equality follows from the induction hypothesis. Now we need to show that $\mathit{dist}_M^{\mathsf{LTL}}(\mu F) = \mu G$. First, $\mathit{dist}_M^{\mathsf{LTL}}(\mu F)$ is a fixpoint of G:

$$\begin{aligned}G(\mathit{dist}_M^{\mathsf{LTL}}(\mu F)) &= \mathit{dist}_M^{\mathsf{LTL}}(|\varphi_2|_M \cup (|\varphi_1|_M \cap M_\circ(\mu F)))\\&= \mathit{dist}_M^{\mathsf{LTL}}(\mu F)\,.\end{aligned}$$

It is also the least fixpoint. Let A be a fixpoint of G. Then $[\![A]\!]_M$ is a fixpoint of F by

$$\begin{aligned}[\![A]\!]_M &= [\![G(A)]\!]_M\\&= |\varphi_2|_M \cup (|\varphi_1|_M \cap M_\circ([\![A]\!]_M)))\\&= F([\![A]\!]_M)\end{aligned}$$

and from μF being the least fixpoint of F it follows that $\mu F \subseteq [\![A]\!]_M$, and therefore $[\![\mu F]\!]_M \subseteq A$.

□

Proof of Lemma 14. $\mathcal{T}^l(M), i \vDash_{\mathsf{LTL}} \varphi$ iff (using Lemma 8) $i \in |\mathsf{L2M}(\varphi)|_{\mathcal{M}^l(\mathcal{T}^l(M))}$ iff (using Lemma 12 and Lemma 13) $\mathit{dist}_M^{\mathsf{LTL}}([\![i]\!]_M^{\mathsf{LTL}}) \in \mathit{dist}_M^{\mathsf{LTL}}(|\mathsf{L2M}(\varphi)|_M)$ iff (since $\mathit{dist}_M^{\mathsf{LTL}}$ is injective by Lemma 12) $[\![i]\!]_M^{\mathsf{LTL}} \in |\mathsf{L2M}(\varphi)|_M$. □

Proof of Theorem 15. The proof goes similarly to the proof of Theorem 9, except that it uses Lemma 14 instead of Lemma 8. □

A.3 HyperLTL

Definition 35 (HyperLTL semantics)**.** *The semantics of HyperLTL is defined w.r.t. a nonempty set of traces T, called a* HyperLTL model*, a valuation $\Pi\colon V \to T$, and a number $i \in \mathbb{N}_{\geq 1}$, as the relation $\vDash_{\mathsf{hLTL}} \subseteq \mathbf{Mod}_{\mathsf{hLTL}} \times (V \to T) \times \mathbb{N}_{\geq 1} \times \mathbf{\Phi}_{\mathsf{hLTL}}$ inductively defined as follows:*

- $T, \Pi, i \vDash_{\mathsf{hLTL}} a_\pi$ *iff* $a \in \Pi(\pi)[i]$;
- $T, \Pi, i \vDash_{\mathsf{hLTL}} \neg\psi$ *iff* $T, \Pi, i \nvDash_{\mathsf{hLTL}} \psi$;
- $T, \Pi, i \vDash_{\mathsf{hLTL}} \psi_1 \vee \psi_2$ *iff* $T, \Pi, i \vDash_{\mathsf{hLTL}} \psi_1$ *or* $T, \Pi, i \vDash_{\mathsf{hLTL}} \psi_2$;
- $T, \Pi, i \vDash_{\mathsf{hLTL}} \circ\psi$ *iff* $T, \Pi, i+1 \vDash_{\mathsf{hLTL}} \psi$;
- $T, \Pi, i \vDash_{\mathsf{hLTL}} \psi_1 \, U \, \psi_2$ *iff there exists* $j \geq i$ *such that* $T, \Pi, j \vDash_{\mathsf{hLTL}} \psi_2$ *and for all* $i \leq k < j$ *we have* $T, \Pi, k \vDash_{\mathsf{hLTL}} \psi_1$;
- $T, \Pi, i \vDash_{\mathsf{hLTL}} \exists\pi \,.\, \varphi$ *iff there exists* $\tau \in T$ *such that* $T, \Pi[\tau/\pi], i \vDash_{\mathsf{hLTL}} \varphi$;
- $T, \Pi, i \vDash_{\mathsf{hLTL}} \forall\pi \,.\, \varphi$ *iff for all* $\tau \in T$ *we have* $T, \Pi[\tau/\pi], i \vDash_{\mathsf{hLTL}} \varphi$;

where $\Pi[\tau/\pi]$ *denotes the valuation* Π' *such that* $\Pi'(\pi) = \tau$ *and* $\Pi'(\pi') = \Pi(\pi)$ *for all* $\pi' \not\equiv \pi$.

A.3.1 From HyperLTL Models to Γ^{hLTL}-models

Definition 36. *Let* $(M, app_M(\_,\_), \{\sigma_M\}_{\sigma\in\Sigma^{\mathsf{LTL}}})$ *be a* Σ^{LTL}*-model. We define*

- *a function* $M_\circ(\_) : M \to \mathcal{P}(M)$ *defined by*

$$M_\circ(m) = appext_M(\circ_M, \{m\})\,;$$

- *a function* $M_{\bar\circ}(\_) : M \to \mathcal{P}(M)$ *defined by*

$$M_{\bar\circ}(m) = appext_M(\bar\circ_M, \{m\})\ ;$$

- *a function* $M_{row}(\_) : M \to \mathcal{P}(M)$ *defined by*

$$M_{row}(m) = appext_M(row_M, \{m\})\ ;$$

- *a function* $M_{col}(\_) : M \to \mathcal{P}(M)$ *defined by*

$$M_{col}(m) = appext_M(col_M, \{m\})\ ;$$

- *a function* $M_{sc}(\_,\_) : M \times M \to \mathcal{P}(M)$ *defined by*

$$M_{sc}(m_1, m_2) = appext_M(appext_M(sc_M, \{m_1\}), \{m_2\})\ ;$$

- *a function* $M_{eq}(\_,\_) : M \times M \to \mathcal{P}(M)$ *defined by*

$$M_{eq}(m_1, m_2) = appext_M(appext_M(eq_M, \{m_1\}), \{m_2\})\ ;$$

- *a set* $M_{Trace} = |[\![Trace]\!]|_M$.
- *a set* $M_{TrSuf} = |[\![TrSuf]\!]|_M$.

When convenient, we use $M_\circ(\_)$, $M_{\bar\circ}(\_)$, $M_{row}(\_)$, $M_{col}(\_)$, $M_{sc}(\_,\_)$, and $M_{eq}(\_,\_)$ to mean their pointwise extensions.

Again, $M_\circ(\_)$ and $M_{\bar\circ}(\_)$ are inversions of each other, because the axiom for $\bar\circ$ is the same as in the LTL case.

Lemma 37. *Let M be a Σ^{LTL}-model. Then $M_\circ(\_)$ and $M_{\bar\circ}(\_)$ are inversions of each other, in the sense that for any $m_1, m_2 \in M$, $m_1 \in M_\circ(m_2)$ if and only if $m_2 \in M_{\bar\circ}(m_1)$.*

They also enjoy the same basic properties, since all the $\circ$-related axioms in Γ^{LTL} are also in Γ^{hLTL}.

Lemma 38. *Let M be a Γ^{hLTL}-model. Then $M_\circ(\_)$ is an injective partial function on M_{TrSuf} (in the sense that for any $m \in M_{TrSuf}$, either $M_\circ(m) = \emptyset$ or $M_\circ(m) = \{m'\}$ for some $m' \in M_{TrSuf}$, and for any $m_1, m_2 \in M_{TrSuf}$, when $M_\circ(m_1) = M_\circ(m_2) = \{m\}$ for some $m \in M$, then $m_1 = m_2$), and $M_{\bar\circ}(\_)$ is a total function on M_{TrSuf} (meaning that for any $m \in M_{TrSuf}$, $M_{\bar\circ}(m) = \{m'\}$ for some $m' \in M_{TrSuf}$) and is injective ($M_{\bar\circ}(m_1) = M_{\bar\circ}(m_2)$ implies $m_1 = m_2$ for any $m_1, m_2 \in M_{TrSuf}$).*

Definition 39. *We define a model translation function $\mathcal{M}^h : \mathbf{Mod}_{\mathsf{hLTL}} \to \mathbf{Mod}_{\mathsf{ML}}(\Gamma^{\mathsf{hLTL}})$, illustrated in Fig. 5, as follows. Let T be a nonempty set of traces, i.e., a HyperLTL model $\emptyset \neq T \subseteq (\mathcal{P}(\mathrm{AP}))^\omega$. We define the carrier set $\mathcal{M}^h(T)$ inductively as the smallest set A such that*

1. *$T \times \mathbb{N}_{\geq 1} \subseteq A$;*
2. *$\{\#\mathsf{def}, \#\mathsf{inh}, \#\mathsf{pair}, \#\mathsf{Trace}, \#\mathsf{TrSuf}, \#\mathsf{next}, \#\mathsf{prev}, \#\mathsf{row}, \#\mathsf{col}, \#\mathsf{sc}, \#\mathsf{eq}\} \subseteq A$;*
3. *$\#\mathsf{sc} \rightsquigarrow (\tau, n) \in A$ for any $(\tau, n) \in T \times \mathbb{N}_{\geq 1}$;*
4. *$\#\mathsf{eq} \rightsquigarrow (\tau, n) \in A$ for any $(\tau, n) \in T \times \mathbb{N}_{\geq 1}$;*
5. *and $\{\#\mathsf{pair} \rightsquigarrow a_1, (\#\mathsf{pair}, a_1, a_2)\} \subseteq A$ for any $a_1, a_2 \in A$*

We let $def_{\mathcal{M}^h(T)} = \{\#\mathsf{def}\}$, $inh_{\mathcal{M}^h(T)} = \{\#\mathsf{inh}\}$, $pair_{\mathcal{M}^h(T)} = \{\#\mathsf{pair}\}$, $Trace_{\mathcal{M}^h(T)} = \{\#\mathsf{Trace}\}$, $TrSuf_{\mathcal{M}^h(T)} = \{\#\mathsf{TrSuf}\}$, $\circ_{\mathcal{M}^h(T)} = \{\#\mathsf{next}\}$, $\bar\circ_{\mathcal{M}^h(T)} = \{\#\mathsf{prev}\}$, $row_{\mathcal{M}^h(T)} = \{\#\mathsf{row}\}$, $col_{\mathcal{M}^h(T)} = \{\#\mathsf{col}\}$, $sc_{\mathcal{M}^h(T)} = \{\#\mathsf{sc}\}$, $eq_{\mathcal{M}^h(T)} = \{\#\mathsf{eq}\}$, and $a_{\mathcal{M}^h(T)} = \{(\tau, n) \mid \in \mathbb{N}_{\geq 1}, \tau \in T, a \in \tau[n]\}$ for any $a \in \mathrm{AP}$, and define the application as follows:

1. *$app_{\mathcal{M}^h(T)}(\#\mathsf{def}, m) = \mathcal{M}^h(T)$ for any $m \in \mathcal{M}^h(T)$;*
2. *$app_{\mathcal{M}^h(T)}(\#\mathsf{inh}, \#\mathsf{Trace}) = \{(\tau, 1) \mid \tau \in T\}$;*
3. *$app_{\mathcal{M}^h(T)}(\#\mathsf{inh}, \#\mathsf{TrSuf}) = T \times \mathbb{N}_{\geq 1}$;*
4. *$app_{\mathcal{M}^h(T)}(\#\mathsf{pair}, m) = \{\#\mathsf{pair} \rightsquigarrow m\}$ for any $m \in \mathcal{M}^h(T)$;*
5. *$app_{\mathcal{M}^h(T)}(\#\mathsf{pair} \rightsquigarrow \mathsf{m}_1, m_2) = (\#\mathsf{pair}, m_1, m_2)$ for any $m_1, m_2 \in \mathcal{M}^h(T)$;*

6. $app_{\mathcal{M}^h(T)}(\#\mathsf{next}, (\tau, 1)) = \emptyset$;

7. $app_{\mathcal{M}^h(T)}(\#\mathsf{next}, (\tau, (n+1))) = \{(\tau, n)\}$ *for all* $(\tau, n) \in T \times \mathbb{N}_{\geq 1}$;

8. $app_{\mathcal{M}^h(T)}(\#\mathsf{prev}, (\tau, n)) = \{(\tau, n+1)\}$ *for all* $(\tau, n) \in T \times \mathbb{N}_{\geq 1}$;

9. $app_{\mathcal{M}^h(T)}(\#\mathsf{sc}, (\tau, n)) = \{\#\mathsf{sc} \rightsquigarrow (\tau, n)\}$ *for all* $(\tau, n) \in T \times \mathbb{N}_{\geq 1}$;

10. $app_{\mathcal{M}^h(T)}(\#\mathsf{sc} \rightsquigarrow (\tau_1, n_1), (\tau_2, n_2)) = \mathcal{M}^h(T)$ *if* $n_1 = n_2$, *otherwise* $\emptyset$, *for all* $(\tau_1, n_1), (\tau_2, n_2) \in T \times \mathbb{N}_{\geq 1}$;

11. $app_{\mathcal{M}^h(T)}(\#\mathsf{eq}, (\tau, n)) = \{\#\mathsf{eq} \rightsquigarrow (\tau, n)\}$ *for all* $(\tau, n) \in T \times \mathbb{N}_{\geq 1}$;

12. $app_{\mathcal{M}^h(T)}(\#\mathsf{eq} \rightsquigarrow (\tau_1, n_1), (\tau_2, n_2)) = \mathcal{M}^h(T)$ *if for all* $n \in \mathbb{N}$ *and for all* $a \in \mathrm{AP}$, $(\tau_1, n_1 + n) \in a_{\mathcal{M}^h(T)}$ *if and only if* $(\tau_2, n_2 + n) \in a_{\mathcal{M}^h(T)}$, *otherwise* $\emptyset$, *for all* $(\tau_1, n_1), (\tau_2, n_2) \in T \times \mathbb{N}_{\geq 1}$;

13. $app_{\mathcal{M}^h(T)}(\#\mathsf{col}, (\tau, n)) = \{(\tau', n) \mid \tau' \in T\}$ *for all* $(\tau, n) \in T \times \mathbb{N}_{\geq 1}$;

14. $app_{\mathcal{M}^h(T)}(\#\mathsf{row}, (\tau, n)) = \{(\tau, n') \mid n' \in \mathbb{N}_{\geq 1}\}$ *for all* $(\tau, n) \in T \times \mathbb{N}_{\geq 1}$;

15. $app_{\mathcal{M}^h(T)}((\#\mathsf{pair}, k_1, v), k_2) = \{v\}$ *if* $k_1 = k_2$, *otherwise* $\emptyset$.

16. *and* $app_{\mathcal{M}^h(T)}(a, b) = \emptyset$ *otherwise.*

Definition 40. *Let* $\rho_T^{\mathsf{hLTL}} : (V \to T) \to (\textsc{Var} \to \mathcal{M}^h(T))$ *denote the function that maps a HyperLTL valuation* $\Pi \colon V \to T$ *to the* $\mathcal{M}^h(T)$*-valuation* $\rho_T^{\mathsf{hLTL}}(\Pi) \colon \textsc{Var} \to \mathcal{M}^h(T)$ *defined as* $\rho_T^{\mathsf{hLTL}}(\Pi)(\pi) = (\Pi(\pi), 1)$ *for all* $\pi \in V$; $\rho_T^{\mathsf{hLTL}}(\Pi)(x) = m$ *for all* $x \in \mathrm{EV} \setminus V$, *where* $m \in \mathcal{M}^h(T)$ *is an arbitrary element; and* $\rho_T^{\mathsf{hLTL}}(\Pi)(x) = \emptyset$ *for all* $X \in \mathrm{SV}$. *The second and third case are present only to have a valid definition; recall that interpretation of variables that are not free in a formula does not influence the formula's semantics.*

Definition 41. *For any* Γ^{hLTL}*-model* M, *we define the function* $M_{\bar{\circ}} : M \to M$ *defined by* $M_{\bar{\circ}}(m) = m'$, *where* m' *is the unique element satisfying* $\{m'\} = appext_M(\bar{\circ}_M, \{m\})$. *We further define the function* $[\![\_]\!]_M^{\mathsf{hLTL}} : \mathbb{N}_{\geq 1} \to \mathcal{P}(M_{TrSuf})$ *as* $[\![0]\!]_M^{\mathsf{hLTL}} = M_{Trace}$ *and* $[\![n+1]\!]_M^{\mathsf{hLTL}} = M_{\bar{\circ}}([\![n]\!]_M^{\mathsf{hLTL}})$ *for all* $n \in \mathbb{N}$. *Intuitively,* $[\![i]\!]_M^{\mathsf{hLTL}}$ *represents the set of all elements of* M *at the* i*th column, i.e., the set of suffixes* $\{\tau[i..] \mid \tau$ *is a (full) trace*$\}$ *(see Fig. 5).*

Lemma 42. *The function* $[\![\_]\!]_M^{\mathsf{hLTL}} : \mathbb{N}_{\geq 1} \to \mathcal{P}(M_{TrSuf})$ *is well-defined, meaning that for any* $i \in \mathbb{N}_{\geq 1}$, $[\![i]\!]_M^{\mathsf{hLTL}} \subseteq M_{TrSuf}$.

Proof of Lemma 42. By induction on i from the fact that $M_{Trace} \subseteq M_{TrSuf}$ (because of the axiom (TraceSuffix)) and Lemma 38. □

Lemma 43. $[\![i]\!]_{\mathcal{M}^h(T)}^{\mathsf{hLTL}} = \{(\tau, i) \mid \tau \in T\}$ *for any HyperLTL model* T.

Proof of Lemma 43. By induction on i. □

Proposition 44. *Function* $\mathcal{M}^h$ *is well-defined, i.e.,* $\mathcal{M}^h(T) \vDash \Gamma^{\mathsf{hLTL}}$ *for any HyperLTL model* T.

Proof of Proposition 44. Let $\rho : V \to \mathcal{M}^h(T)$ be a valuation. We will prove the axioms one by one.

- (DEFINEDNESS) - $|\lceil x \rceil|_\rho = appext_{\mathcal{M}^h(T)}(\{\#\mathsf{def}\}, |x|_\rho) = app_{\mathcal{M}^h(T)}(\#\mathsf{def}, \{\rho(x)\}) = app_{\mathcal{M}^h(T)}(\#\mathsf{def}, \rho(x)) = \mathcal{M}^h(T)$

- (PAIRFUNCTION) - Let $m_1, m_2 \in \mathcal{M}^h(T)$. Then $|\langle x, y\rangle|_{\rho[m_1/x][m_2/y} = |pair\ x\ y|_{\rho[m_1/x][m_2/y]} = appext_{\mathcal{M}^h(T)}(|pair\ x|_{\rho[m_1/x][m_2/y]}, \{\rho[m_1/x][m_2/y](y)\}) = appext_{\mathcal{M}^h(T)}(\{\#\mathsf{pair} \rightsquigarrow m_1\}, \{m_2\}) = \{(\#\mathsf{pair}, \mathsf{m_1}, \mathsf{m_2})\}$ which is a singleton set.

- (PAIRINJECTIVE) - Follows by simple computation and by injectivity of (meta)-tuples.

- (PROJECTPAIR1) - Follows by simple computation.

- (PROJECTPAIR2) - Follows by simple computation.

- (KEYVALUE) - If $\rho(k_1) = \rho(k_2)$, then

$$\begin{aligned} |\langle k_1, v\rangle\ k_2|_\rho &= app_{\mathcal{M}^h(T)}((\#\mathsf{pair}, \rho(\mathsf{k_1}), \rho(\mathsf{v})), \rho(k_2)) \\ &= \{\rho(v)\} \\ &= \mathcal{M}^h(T) \cap \{\rho(v)\} \\ &= |(k_1 = k_2) \wedge v|_\rho\,. \end{aligned}$$

  Otherwise, $|\langle k_1, v\rangle\ k_2|_\rho = app_{\mathcal{M}^h(T)}((\#\mathsf{pair}, \rho(\mathsf{k_1}), \rho(\mathsf{v})), \rho(k_2)) = \emptyset = \emptyset \cap \{\rho(v)\} = |(k_1 = k_2) \wedge v|_\rho$.

- (TRACESUFFIX) - We need to show that $\mathcal{M}^h(T)_{TrSuf} = T \times \mathbb{N}_{\geq 1}$ is the least fixpoint of the function $F(A) = \mathcal{M}^h(T)_{TrSuf} \cup appext_{\mathcal{M}^h(T)}(\{\#\mathsf{prev}\}, A)$. That it is a fixpoint follows from a straightforward computation, while for the minimality we assume that A is a fixpoint of F and prove that for any $n \in \mathbb{N}_{\geq 1}$ and $\tau \in T$, $(\tau, n) \in A$, by induction on n.

- (INF) - We need to show that for any $(\tau_1, n_2) \in \mathcal{M}^h(T)_{TrSuf}$ there exists some $(\tau_2, n_2) \in \mathcal{M}^h(T)_{TrSuf}$ such that $(\tau_1, n_1) \in app_{\mathcal{M}^h(T)}(\#\mathsf{next}, (\tau_2, n_2))$; but that holds by the choice $(\tau_2, n_2) = (\tau_1, n_1 + 1)$.

- (NEXTOUT) - To show that $|\circ(\neg[\![TrSuf]\!])|_\rho \subseteq |\neg[\![TrSuf]\!]|_\rho$, it is enough to show that $|\circ(\neg[\![TrSuf]\!])|_\rho = appext_{\mathcal{M}^h(T)}(\{\#\mathsf{next}\}, \mathcal{M}^h(T) \setminus T \times \mathbb{N}_{\geq 1}) = \emptyset$, which holds because for any $m \in \mathcal{M}^h(T) \setminus T \times \mathbb{N}_{\geq 1}$, $app_{\mathcal{M}^h(T)}(\#\mathsf{next}, m) = \emptyset$.

- (NEXTPFUN) - holds because $app_{\mathcal{M}^h(T)}(\#\mathsf{next}, (\tau, n))$ is either empty set or a singleton set for any $(\tau, n) \in T \times \mathbb{N}_{\geq 1}$.

- (NEXTINJ) - Let $(\tau_1, n_1), (\tau_2, n_2) \in T \times \mathbb{N}_{\geq 1}$ be such that

$$app_{\mathcal{M}^h(T)}(\#\mathsf{next}, (\tau_1, n_1)) = app_{\mathcal{M}^h(T)}(\#\mathsf{next}, (\tau_2, n_2)) \neq \emptyset\,.$$

Then $n_1 = n_1' + 1$ and $n_2 = n_2' + 1$ for some $n_1', n_2' \in \mathbb{N}_{\geq 1}$, and $\{(\tau_1, n_1')\} = app_{\mathcal{M}^h(T)}(\#\mathsf{next}, (\tau_1, n_1' + 1)) = app_{\mathcal{M}^h(T)}(\#\mathsf{next}, (\tau_2, n_2' + 1)) = \{(\tau_2, n_2')\}$. Therefore, $(\tau_1, n_1) = (\tau_1, n_1' + 1) = (\tau_2, n_2' + 1) = (\tau_2, n_2)$.

- (Prev) - Let $\rho\,(x) = (\tau, i)$. Then

$$\begin{aligned}\bar{\rho}\,(\exists y.y \wedge (x \in \circ y)) & \\ &= \{(\tau', i') \mid (\tau, i)\,\mathcal{M}^h(T)_\circ((\tau', i'))\} \\ &= \{(\tau, i+1)\} \\ &= \bar{\rho}\,(\bar{\circ}x)\ .\end{aligned}$$

- (AtomicProp) - follows by construction.

- (Row) - Let $(\tau, i) \in T \times \mathbb{N}_{\geq 1}$. We need to show that $|row(x)|_{\rho[(\tau,i)/x]} = \{(\tau, i') \mid i' \in \mathbb{N}_{\geq 1}\}$ is the least fixpoint of the function $F(A) = \{(\tau, i)\} \cup \mathcal{M}^h(T)_\circ(A) \cup \mathcal{M}^h(T)_\circ(A)$. It is indeed a fixpoint:

$$\begin{aligned}&F(\{(\tau, i') \mid i' \in \mathbb{N}_{\geq 1}\}) \\ &= \{(\tau, i)\} \cup \{(\tau, i' - 1) \mid i' \in \mathbb{N}_{\geq 1}, i' > 1\} \\ &\cup \{(\tau, i' + 1) \mid i' \in \mathbb{N}_{\geq 1}\} \\ &= \{(\tau, i') \mid i' \in \mathbb{N}_{\geq 1}\}.\end{aligned}$$

  It is also the least fixpoint. Let A be some fixpoint of F. We will show that $\{(\tau, i') \mid i' \in \mathbb{N}_{\geq 1}\} \subseteq A$, which is equivalent to $\forall j \in \mathbb{N}_0, k \in \mathbb{N}_{\geq 1}.\, i - j \leq k \leq i + j \implies (\tau, k) \in F(A) = A$. We will proceed by induction on j.

  - For $j = 0$, $(\tau, i) \in F(A)$.
  - For $j > 0$, we assume the induction hypothesis (IH) $\forall k \in \mathbb{N}_{\geq 1}.\, i - j + 1 \leq k \leq i + j - 1 \implies (\tau, k) \in A$, and need to show that $(\tau, i + j) \in F(A)$ and if $i - j > 1$, then $(\tau, i - j) \in F(A)$. For the former, it follows from IH that $(\tau, i + j - 1) \in A$, therefore $(\tau, i + j) \in \bar{\circ}_{\mathcal{M}^h(T)}(A)$, therefore $(\tau, i + j) \in F\,(A)$. For the latter, assuming $i - j > 1$ we have $(\tau, i - j + 1) \in A$, therefore $(\tau, i - j) \in \mathcal{M}^h(T)_\circ(A)$, therefore $(\tau, i - j) \in F\,(A)$.

- (SC) - As the axiom desugars to

$$\begin{aligned}&\forall x, y\colon \mathit{TrSuf}.\, \mathit{sc}\ x\ y = \\ &(\mu\ \mathit{sc}\,.\,\exists x.\,\exists y.\,\langle x, \langle y, (x \in [\![\mathit{Trace}]\!] \wedge y \in [\![\mathit{Trace}]\!]) \vee \\ &\mathit{sc}\ (\circ x)\ (\circ y)\rangle\rangle)\ x\ y\,,\end{aligned}$$

  we will prove that

$$\begin{aligned}&|\mu\ \mathit{sc}.\,\exists x.\,\exists y.\,\langle x, \langle y, (x \in [\![\mathit{Trace}]\!] \wedge y \in [\![\mathit{Trace}]\!]) \vee \\ &\mathit{sc}\ (\circ x)\ (\circ y)\rangle\rangle|_\rho = \mathit{sc}'_{\mathcal{M}^h(T)}\end{aligned}$$

where $sc'_{\mathcal{M}^h(T)} = \{(\#\mathsf{pair}, (\tau, i), (\#\mathsf{pair}, (\tau', i), \gamma)) \mid \tau, \tau' \in T, i \in \mathbb{N}_{\geq 1}, \gamma \in \mathcal{M}^h(T)\}$, from which the desired equality directly follows. That means proving that $sc'_{\mathcal{M}^h(T)}$ is the least fixpoint of the function F, defined as

$$
\begin{aligned}
F(A) = \exists x.\, \exists y.\, \langle x, \langle y, (x \in [\![\mathit{Trace}]\!] \wedge y \in [\![\mathit{Trace}]\!]) \\
&\vee sc\ (\circ x)\ (\circ y)\rangle\rangle|_{\rho[A/sc]} \\
&= \bigcup_{\alpha,\beta \in \mathcal{M}^h(T)} \{(\#\mathsf{pair}, \alpha, (\#\mathsf{pair}, \beta, \gamma \mid \\
&\gamma \in \left(\lceil \alpha \cap \mathcal{M}^h(T)_{\mathit{Trace}} \rceil \cap \lceil \beta \cap \mathcal{M}^h(T)_{\mathit{Trace}} \rceil\right)\} \\
&\cup A\left(\mathcal{M}^h(T)_\circ(\alpha), \mathcal{M}^h(T)_\circ(\beta)\right))) \\
&= (\bigcup_{\alpha,\beta \in \mathcal{M}^h(T)_{\mathit{Trace}}} \{(\#\mathsf{pair}, \alpha, (\#\mathsf{pair}, \beta, \gamma)) \mid \\
&\gamma \in \mathcal{M}^h(T)\}) \\
&\cup (\bigcup_{\alpha,\beta \in \mathcal{M}^h(T)} (\#\mathsf{pair}, \alpha, (\#\mathsf{pair}, \beta, \\
&(A\ (\mathcal{M}^h(T)_\circ(\alpha))\ (\mathcal{M}^h(T)_\circ(\beta)))))) \\
&= \{(\#\mathsf{pair}, (\tau_1, 1), (\#\mathsf{pair}, (\tau_2, 1), \gamma)) \mid \\
&\tau_1, \tau_2 \in T, \gamma \in \mathcal{M}^h(T)\} \\
&\cup \{(\#\mathsf{pair}, (\tau_1, i_1 + 1), (\#\mathsf{pair}, (\tau_2, i_2 + 1), \gamma)) \mid \\
&(\#\mathsf{pair}, (\tau_1, i_1), (\#\mathsf{pair}, (\tau_2, i_2), \gamma)) \in A\}\,.
\end{aligned}
$$

Straightforward computation yields that $sc'_{\mathcal{M}^h(T)}$ is inded a fixpoint of F. It is also a least fixpoint: if A is a fixpoint of F, then $sc'_{\mathcal{M}^h(T)} \subseteq A$. Equivalently,

$$
\begin{aligned}
&\forall i \in \mathbb{N}_{\geq 1}.\, \forall \tau_1, \tau_2 \in T.\, \forall \gamma \in \mathcal{M}^h(T). \\
&(\#\mathsf{pair}, (\tau_1, i), (\#\mathsf{pair}, (\tau_2, i), \gamma)) \in F(A)\ ,
\end{aligned}
$$

which can be easily proved by induction on i.

- (Col) - Let $(\tau, i) \in T \times \mathbb{N}_{\geq 1}$. Then

$$
\begin{aligned}
&|\exists y\colon \mathit{TrSuf}.\, y \wedge (sc\ x\ y)|_{\rho[(\tau,i)/x]} \\
&= \bigcup_{(\tau',i') \in \mathcal{M}^h(T)_{\mathit{TrSuf}}} \{(\tau', i')\} \cap \mathcal{M}^h(T)_{sc}((\tau, i), (\tau', i')) \\
&= \{(\tau', i) \mid \tau' \in T\} \\
&= \mathcal{M}^h(T)_{col}((\tau, i))\,.
\end{aligned}
$$

- (EQ) - As the axiom desugars to

$$\begin{aligned}&\forall x, y : \mathit{TrSuf}.\ \mathit{eq}\ x\ y = \\ &(\nu\ \mathit{eq}.\ \exists x.\ \exists y.\ \langle x, \langle y, (\bigwedge_{a \in \mathrm{AP}} x \in a \leftrightarrow y \in a) \\ &\wedge\ \mathit{eq}\ (\bar{\circ} x)(\bar{\circ} y)\rangle\rangle)\ x\ y\,,\end{aligned}$$

we will prove that

$$\begin{aligned}&|\nu\ \mathit{eq}.\ \exists x.\ \exists y.\ \langle x, \langle y, (\bigwedge_{a \in \mathrm{AP}} x \in a \leftrightarrow y \in a) \\ &\wedge\ \mathit{eq}\ (\bar{\circ} x)(\bar{\circ} y)\rangle\rangle|_\rho = \mathit{eq}'_{\mathcal{M}^h(T)}\end{aligned}$$

where

$$\mathit{eq}' = \{(\#\mathsf{pair}, m_1, (\#\mathsf{pair}, \mathsf{m}_2, \gamma)) \mid \gamma \in \mathcal{M}^h(T) \wedge \mathcal{M}^h(T) = \mathcal{M}^h(T)_{\mathit{eq}}(m_1, m_2)\}\,.$$

That means proving that eq' is the greatest fixpoint of the function F, defined by

$$\begin{aligned}F(\mathit{eq}) = \bigcup_{m_1, m_2 \in M} &\{(\#\mathsf{pair}, m_1, (\#\mathsf{pair}, \mathsf{m}_2, \gamma)) \mid \\ &\gamma \in (\bigcap_{a \in AP} \mathcal{M}^h(T) \setminus \\ &((m_1 \in_{\mathcal{M}^h(T)} a_{\mathcal{M}^h(T)}) \Delta (m_2 \in_{\mathcal{M}^h(T)} a_{\mathcal{M}^h(T)}))) \\ &\cap\ \mathit{eq}\ (\bar{\circ}_{\mathcal{M}^h(T)}(m_1))(\bar{\circ}_{\mathcal{M}^h(T)}(m_2))\}\end{aligned}$$

where by $m \in_M S$ we mean $\lceil \{m\} \cap S \rceil_M$. First, $F(\mathit{eq}') = \mathit{eq}'$. Consider $m_1, m_2, m \in \mathcal{M}^h(T)$. Then $(\#\mathsf{pair}, \mathsf{m}_1, (\#\mathsf{pair}, \mathsf{m}_2, \mathsf{m})) \in \mathit{eq}'$ if and only if there exists some $(\tau_1, i_1), (\tau_2, i_2) \in T \times \mathbb{N}_{\geq 1}$ such that $m_1 = (\tau_1, i_1)$, $m_2 = (\tau_2, i_2)$, and $\forall a \in \mathrm{AP}. \forall n \in \mathbb{N}. (\tau_1, i_1 + n) \in a_{\mathcal{M}^h(T)} \leftrightarrow (\tau_2, i_2 + n) \in a_{\mathcal{M}^h(T)}$; if and only if $\forall a \in \mathrm{AP}. (\tau_1, i_1) \in a_{\mathcal{M}^h(T)} \leftrightarrow (\tau_2, i_2) \in a_{\mathcal{M}^h(T)}$ and $\forall a \in \mathrm{AP}. \forall n \in \mathbb{N}. (\tau_1, i_1 + 1 + n) \in a_{\mathcal{M}^h(T)} \leftrightarrow (\tau_2, i_2 + 1 + n) \in a_{\mathcal{M}^h(T)}$; if and only if $m \in \bigcap_{a \in \mathrm{AP}} \mathcal{M}^h(T) \setminus (((\tau_1, i_1) \in_{\mathcal{M}^h(T)} a_{\mathcal{M}^h(T)}) \Delta ((\tau_2, i_2) \in_{\mathcal{M}^h(T)} a_{\mathcal{M}^h(T)}))$ and $(\#\mathsf{pair}, (\tau_1, i_1{+}1), (\#\mathsf{pair}, (\tau_2, i_2{+}1), m)) \in \mathit{eq}'$, if and only if $(\#\mathsf{pair}, (\tau_1, i_2), (\#\mathsf{pair}, (\tau_2, i_2), m)) \in F(\mathit{eq}')$. Next we need to show it is the greatest fixpoint. Given $A = F(A)$, we will show that $A \subseteq \mathit{eq}'$. Since A is a fixpoint of F, it follows that for any $m_1, m_2, m \in \mathcal{M}^h(T)$ such that $(\#\mathsf{pair}, m_1, (\#\mathsf{pair}, m_2, m)) \in A$ there exists some $(\tau_1, i_1), (\tau_2, i_2) \in T \times \mathbb{N}_{\geq 1}$ such that $m_1 = (\tau_1, i_1)$ and $m_2 = (\tau_2, i_2)$, because for any other element m' we have $\mathcal{M}^h(T)_{\bar{\circ}}(m) = \emptyset$; therefore, we do not have to consider those elements when showing the inclusion. Now, let $(\tau_1, i_1), (\tau_2, i_2) \in T \times \mathbb{N}_{\geq 1}$ and $m \in \mathcal{M}^h(T)$ be such that $(\#\mathsf{pair}, (\tau_1, i_1), (\#\mathsf{pair}, (\tau_2, i_2), m)) \notin \mathit{eq}'_M$. Then there exists some $a \in \mathrm{AP}$ and $n \in \mathbb{N}$ such that $(\tau_1, i_1 + n) \in a_{\mathcal{M}^h(T)} \not\leftrightarrow (\tau_2, i_2 + n) \in a_{\mathcal{M}^h(T)}$. We will show that $\forall j. 0 \leq j \leq n \implies (\#\mathsf{pair}, (\tau_1, i_1{+}n{-}j), (\#\mathsf{pair}, (\tau_2, i_2{+}n{-}$

$j), m)) \notin A$, from which it follows that $(\#\mathsf{pair}, (\tau_1, i_1), (\#\mathsf{pair}, (\tau_2, i_2), m)) \notin A$ by choosing n for j. We will proceed by induction on j.

  - $i = 0$ - we have $\bigcap_{a \in \mathrm{AP}} \mathcal{M}^h(T) \backslash (((\tau_1, i_1 + n) \in_{\mathcal{M}^h(T)} a_{\mathcal{M}^h(T)}) \Delta ((\tau_2, i_2) \in_{\mathcal{M}^h(T)} a_{\mathcal{M}^h(T)})) = \emptyset$, therefore $(\#\mathsf{pair}, (\tau_1, i_1 + n), (\#\mathsf{pair}, (\tau_2, i_2 + n), m)) \notin F(A) = A$.
  - $i > 0$ - assuming $(\#\mathsf{pair}, (\tau_1, i_1 + n - (j-1)), (\#\mathsf{pair}, (\tau_2, i_2 + n - (j-1)), m)) \notin A$, it follows that $m \notin A(\bar{\circ}_{\mathcal{M}^h(T)}((\tau_1, i_1 + n - j)), \bar{\circ}_{\mathcal{M}^h(T)}((\tau_2, i_2 + n - j)))$, and therefore $(\#\mathsf{pair}, (\tau_1, i_1 + n - j), (\#\mathsf{pair}, (\tau_2, i_2 + n - j), m)) \notin F(A) = A$.

- (Set) - Let $\tau_1, \tau_2 \in T$. Assuming $|eq\ x\ y|_{\rho[(\tau_1,1)/x][(\tau_2,1)/y]} \neq \emptyset$, we prove that $|x = y|_{\rho[(\tau_1,1)/x][(\tau_2,1)/y]} = \mathcal{M}^h(T)$; i.e., that $\tau_1 = \tau_2$. From $\mathcal{M}^h(T)_{eq}((\tau_1, 0), (\tau_2, 0)) = \mathcal{M}^h(T)$ we have that for all $a \in \mathrm{AP}$ and for all $n \in \mathbb{N}_{\geq 1}$, $a \in \tau_1[n]$ iff $(\tau_1, n) \in a_{\mathcal{M}^h(T)}$ iff $(\tau_2, n) \in a_{\mathcal{M}^h(T)}$ iff $a \in \tau_2[n]$. Therefore $\tau_1 = \tau_2$.

□

Lemma 45. *Let T be a HyperLTL model. Then for any T-valuation Π, the $\mathcal{M}^h(T)$-valuation $\rho_T^{\mathsf{hLTL}}(\Pi)$ is well-sorted. Moreover, for any well-sorted $\mathcal{M}^h(T)$-valuation ρ there exists a HyperLTL T-valuation Π' such that for any HyperLTL formula φ, $|\mathsf{H2M}(\varphi)|_\rho = |\mathsf{H2M}(\varphi)|_{\rho_T^{\mathsf{hLTL}}(\Pi')}$.*

Proof of Lemma 45. The first part follows directly by the construction of $\rho_T^{\mathsf{hLTL}}(\Pi)$. For the second part, from ρ being well-sorted follows that $\rho(\pi) \in T \times \mathbb{N}_{\geq 1}$ for any $\pi \in V$. We define $\Pi'(\pi) = \tau$ whenever $\rho(\pi) = (\tau, n)$. Then for any $\pi \in V$, $\rho_T^{\mathsf{hLTL}}(\pi) = \rho(\pi)$, and since $\mathrm{FV}(\mathsf{H2M}(\varphi)) \subseteq V$, it follows by Lemma **??**, that $|\mathsf{H2M}(\varphi)|_\rho = |\mathsf{H2M}(\varphi)|_{\rho_T^{\mathsf{hLTL}}(\Pi')}$. □

Lemma 46. *In any Γ^{hLTL}-model M, the Σ^{hLTL} pattern representing a HyperLTL formula φ matches only trace suffixes: for any well-sorted valuation $\rho : \mathrm{VAR} \to M$, $|\mathsf{H2M}(\varphi)|_\rho \subseteq M_{TrSuf}$.*

Proof of Lemma 46. By structural induction on φ. The inductive cases are similar to the proof of Lemma 7, except the cases for quantifier, which easily follow from the induction hypothesis and Lemma 45. It remains to prove the base case. Since a_π desugars to $col(a \wedge row(\pi))$, we need to show that $|col(a \wedge row(\pi))|_\rho \subseteq M_{TrSuf}$. Because of the "set comprehension" structure of the axiom (Col), $M_{col}(A) \subseteq M_{TrSuf}$ whenever $A \subseteq M_{TrSuf}$. But $a \wedge row\pi \subseteq M_{TrSuf}$ whenever $a \subseteq M_{TrSuf}$, which follows by the axiom (AtomicProp).

□

Proof of Lemma 17. Since T is nonempty, also $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)}$ is nonempty; therefore, $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\varphi)|_{\rho_T^{\mathsf{hLTL}}(\Pi)}$ implies $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\mathsf{H2M}(\varphi)|_{\rho_T^{\mathsf{hLTL}}(\Pi)} \neq \emptyset$. For this reason it is sufficient to prove just two implications: that $T, \Pi, i \vDash_{\mathsf{hLTL}} \varphi$ implies $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\varphi)|_{\rho_T^{\mathsf{hLTL}}(\Pi)}$ and that $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\mathsf{H2M}(\varphi)|_{\rho_T^{\mathsf{hLTL}}(\Pi)} \neq \emptyset$ implies $T, \Pi, i \vDash_{\mathsf{hLTL}} \varphi$. We will proceed by structural induction on φ.

- $\varphi \equiv a_\pi$ - Assuming $T, \Pi, i \vDash_{\mathsf{hLTL}} a_\pi$, we prove $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |a_\pi|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$. We know that $a \in \Pi(\pi)(i)$. Therefore, $\{(\Pi(\pi), i)\} \subseteq \{(\Pi(\pi), i') \mid a \in \Pi(\pi)(i'), i' \in \mathbb{N}\}$. Therefore,

$$\begin{aligned}
|a_\pi|_{\rho^{\mathsf{hLTL}}_T(\Pi)} &= |col(a \wedge row(\pi))|_{\rho^{\mathsf{hLTL}}_T(\Pi)} \\
&= col_{\mathcal{M}^h(T)}(a_{\mathcal{M}^h(T)} \cap row_{\mathcal{M}^h(T)}(\Pi(\pi))) \\
&= col_{\mathcal{M}^h(T)}(\{(\Pi(\pi), i') \mid a \in \Pi(\pi)(i'), i' \in \mathbb{N}\}) \\
&\supseteq col_{\mathcal{M}^h(T)}(\{(\Pi(\pi), i)\}) \\
&= [\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)}.
\end{aligned}$$

  For the other implication, assume $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |a_\pi|_{\rho^{\mathsf{hLTL}}_T(\Pi)} \neq \emptyset$. Then there exists some $\tau \in T$ such that $(\tau, i) \in |a_\pi|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \{(\tau', i') \mid \tau' \in T, i' \in \mathbb{N}, a \in \Pi(\pi)(i')\}$. But then $a \in \Pi(\pi)(i)$, and $T, \Pi, i \vDash a_\pi$.

- $\varphi \equiv \neg\varphi'$ - Assume $T, \Pi, i \vDash_{\mathsf{hLTL}} \neg\varphi'$. Then $T, \Pi, i \nvDash_{\mathsf{hLTL}} \varphi'$, and from the induction hypothesis $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\mathsf{H2M}(\varphi')|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \emptyset$. Since $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq M_{TrSuf} \subseteq M$ (Lemma 42), also $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq \mathcal{M}^h(T)_{TrSuf} \cap (\mathcal{M}^h(T) \setminus |\mathsf{H2M}(\varphi')|_{\rho^{\mathsf{hLTL}}_T(\Pi)}) = |[\![TrSuf]\!] \wedge \neg\mathsf{H2M}(\varphi')|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$. For the second implication, let there be some $\tau \in T$ such that $(\tau, i) \in |[\![TrSuf]\!] \wedge \neg\varphi'|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \mathcal{M}^h(T)_{TrSuf} \cap (\mathcal{M}^h(T) \setminus |\mathsf{WellSorted}(\varphi') \rightarrow [\![Trace]\!] \subseteq \mathsf{H2M}(\varphi')|_{\rho^{\mathsf{hLTL}}_T(\Pi)})$. Then $(\tau, i) \notin |\mathsf{WellSorted}(\varphi') \rightarrow [\![Trace]\!] \subseteq \mathsf{H2M}(\varphi')|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$, therefore $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \not\subseteq |\mathsf{WellSorted}(\varphi') \rightarrow [\![Trace]\!] \subseteq \mathsf{H2M}(\varphi')|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$, and from the induction hypothesis $T, \Pi, i \nvDash_{\mathsf{hLTL}} \varphi'$. Therefore, $T, \Pi, i \vDash_{\mathsf{hLTL}} \neg\varphi'$.

- $\varphi \equiv \psi_1 \vee \psi_2$ - If $T, \Pi, i \vDash_{\mathsf{hLTL}} \psi_1 \vee \psi_2$, then $T, \Pi, i \vDash_{\mathsf{hLTL}} \psi_j$ for some $j \in \{1, 2\}$. But the from the induction hypothesis, $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\psi_j|_{\rho^{\mathsf{hLTL}}_T(\Pi)} \subseteq |\psi_1|_{\rho^{\mathsf{hLTL}}_T(\Pi)} \cup |\psi_2|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = |\psi_1 \vee \psi_2|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$. On the other hand, if $(\tau, i) \in [\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\psi_1 \vee \psi_2|_{\rho^{\mathsf{hLTL}}_T(\Pi)} \subseteq |\psi_1|_{\rho^{\mathsf{hLTL}}_T(\Pi)} \cup |\psi_2|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$ for some $\tau \in T$, then $(\tau, i) \in [\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\psi_j|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$ for some $j \in \{1, 2\}$. Then from the induction hypothesis $T, \Pi, i \vDash_{\mathsf{hLTL}} \psi_j$, therefore $T, \Pi, i \vDash_{\mathsf{hLTL}} \psi_1 \vee \psi_2$.

- $\varphi \equiv \circ\psi$ - If $T, \Pi, i \vDash_{\mathsf{hLTL}} \circ\psi$, then $T, \Pi, i+1 \vDash_{\mathsf{hLTL}} \psi$, and by the induction hypothesis, $[\![i+1]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$. By Lemma 43, $\{(\tau, i+1) \mid \tau \in T\} \subseteq |\mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$, and by the construction of $\mathcal{M}^h(T)$, $\{(\tau, i) \mid \tau \in T\} \subseteq appext_{\mathcal{M}^h(T)}(\{\#\mathsf{next}\}, |\mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}) = |\circ\mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$. Therefore by Lemma 43, $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\circ\mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$.

  For the other implication, we use Lemma 46 and let $(\tau, i) \in |\circ\mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = appext_{\mathcal{M}^h(T)}(\{\#\mathsf{next}\}, |\mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)})$. By the construction of $\mathcal{M}^h(T)$, we have $(i+1, \tau) \in |\mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$, by the induction hypothesis $T, \Pi, i+1 \vDash_{\mathsf{hLTL}} \psi$, and $T, \Pi, i \vDash_{\mathsf{hLTL}} \circ\psi$ by the definition of $\vDash_{\mathsf{hLTL}}$.

- $\varphi \equiv \psi_1 U \psi_2$ - Since $\psi_1 U \psi_2$ is an alias for $\mu X.\psi_2 \vee (\psi_1 \wedge \circ X)$, in ML we have that $|\psi_1\, U\, \psi_2|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \mu F_U$, where $F_U(A) = |\psi_2|_{\rho^{\mathsf{hLTL}}_T(\Pi)} \cup (|\psi_1|_{\rho^{\mathsf{hLTL}}_T(\Pi)} \cap$

$\mathcal{M}^h(T)_\circ(A)$). Because $\mu F_U = F_U(\mu F_U)$, i.e. μF_U is a fixpoint of F_U, we can expand

$$\begin{aligned} |\psi_1 \, U \, \psi_2|_{\rho_T^{\mathsf{hLTL}}(\Pi)} = |\psi_2|_{\rho_T^{\mathsf{hLTL}}(\Pi)} \cup \\ (|\psi_1|_{\rho_T^{\mathsf{hLTL}}(\Pi)} \cap \mathcal{M}^h(T)_\circ(|\psi_1 \, U \, \psi_2|_{\rho_T^{\mathsf{hLTL}}(\Pi)}) \, . \end{aligned} \tag{4}$$

For $T, \Pi, i \vDash_{\mathsf{hLTL}} \psi_1 U \psi_2$ implies $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi_1) \, U \, \mathsf{H2M}(\psi_2)|_{\rho_T^{\mathsf{hLTL}}(\Pi)}$, we first prove a stronger statement:

Claim 47. *For all $m, n \in \mathbb{N}_{\geq 1}$ such that $n \leq m$,*

$$\begin{aligned} &(T, \Pi, m \vDash_{\mathsf{hLTL}} \psi_2 \wedge \forall o \in \mathbb{N}. \, o < n \implies \\ &T, \Pi, m - o - 1 \vDash_{\mathsf{hLTL}} \psi_1) \implies \\ &\forall p \in \mathbb{N}. \, p \leq n \implies \\ &[\![m - p]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi_1) \, U \, \mathsf{H2M}(\psi_2)|_{\rho_T^{\mathsf{hLTL}}(\Pi)} \, . \end{aligned}$$

Proof. By induction on n.

- $n = 1$ - Assuming $T, \Pi, m \vDash_{\mathsf{hLTL}} \psi_2$, by the (outer) induction hypothesis $[\![m]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi_2)|_{\rho_T^{\mathsf{hLTL}}(\Pi)}$, and from (4) it follows that $[\![m - 0]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi_1) \, U \, \mathsf{H2M}(\psi_2)|_{\rho_T^{\mathsf{hLTL}}(\Pi)}$.
- $n > 1$ - Assuming the induction hypothesis

  $$\begin{aligned} &(T, \Pi, m \vDash_{\mathsf{hLTL}} \psi_2 \wedge \forall o \in \mathbb{N}. \, o < n - 1 \implies \\ &T, \Pi, m - o - 1 \vDash_{\mathsf{hLTL}} \psi_1) \implies \\ &\forall p \in \mathbb{N}. \, p \leq n \implies \\ &[\![m - p]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi_1) \, U \, \mathsf{H2M}(\psi_2)|_{\rho_T^{\mathsf{hLTL}}(\Pi)} \end{aligned}$$

  and assuming $T, \Pi, m \vDash_{\mathsf{hLTL}} \psi_2$ and $\forall o \in \mathbb{N}. \, o < n \implies T, \Pi, m - o - 1 \vDash_{\mathsf{hLTL}} \psi_2$ it follows that $\forall p \in \mathbb{N}. \, p \leq n \implies [\![m - p]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\psi_1 \, U \, \psi_2|_{\rho_T^{\mathsf{hLTL}}(\Pi)}$. It remains to prove the case when $p = n$, i.e. $[\![m - n]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\psi_1 \, U \, \psi_2|_{\rho_T^{\mathsf{hLTL}}(\Pi)}$. Because of the equation (4) it is enough to prove that $[\![m - n]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi_1)|_{\rho_T^{\mathsf{hLTL}}(\Pi)}$ and $[\![m - n]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq \mathcal{M}^h(T)_\circ(|\mathsf{H2M}(\psi_1) \, U \, \mathsf{H2M}(\psi_2)|_{\rho_T^{\mathsf{hLTL}}(\Pi)})$. For the former, we use the outer induction hypothesis and the assumption $T, \Pi, m - (n - 1) - 1 \vDash_{\mathsf{hLTL}} \psi_1$, while the latter holds because by the definition of $\mathcal{M}^h(T)_\circ(\_)$ and because by the (inner) induction hypothesis, $[\![m - (n - 1)]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi_1) \, U \, \mathsf{H2M}(\psi_2)|_{\rho_T^{\mathsf{hLTL}}(\Pi)}$.

□

Now assume $T, \Pi, i \vDash_{\mathsf{hLTL}} \psi_1 \, U \, \psi_2$. By the definition of $\vDash_{\mathsf{hLTL}}$, there exists $j \geq i$ such that $T, \Pi, j \vDash_{\mathsf{hLTL}} \psi_2$ and for all $i \leq k < j$ we have $T, \Pi, k \vDash_{\mathsf{hLTL}}$

ψ_1. When in the above claim we choose m to be j, n to be $j-i$ and p to be n, we get $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} = [\![j-(j-i)]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi_1)\, U\, \mathsf{H2M}(\psi_2)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$.

For $T, \Pi, i \not\models_{\mathsf{hLTL}} \psi_1 U \psi_2$ implies $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\mathsf{H2M}(\psi_1)\, U\, \mathsf{H2M}(\psi_2)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \emptyset$, we first prove a stronger statement:

Claim 48. *For all $m, n \in \mathbb{N}_{\geq 1}$ such that $n \leq m$,*

$$\begin{aligned}
&(T, \Pi, m \not\models_{\mathsf{hLTL}} \psi_1 \wedge \forall o \in \mathbb{N}.\, o \leq n \implies \\
&T, \Pi, m-o \not\models_{\mathsf{hLTL}} \psi_2) \implies \\
&\forall p \in \mathbb{N}.\, p \leq n \implies \\
&[\![m-p]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\mathsf{H2M}(\psi_1)\, U\, \mathsf{H2M}(\psi_2)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \emptyset\,.
\end{aligned}$$

Proof. By induction on n.

- $n = 1$ - Assuming $T, \Pi, m \not\models_{\mathsf{hLTL}} \psi_1$ and $T, \Pi, m-0 \not\models_{\mathsf{hLTL}} \psi_1$, by the (outer) induction hypothesis it holds that $|\mathsf{H2M}(\psi_1)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} \cap [\![m]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} = \emptyset$ and $|\mathsf{H2M}(\psi_2)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} \cap [\![m]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} = \emptyset$. But then by (4), $\emptyset = |\mathsf{H2M}(\psi_1)\, U\, \mathsf{H2M}(\psi_2)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} \cap [\![m]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)}$.
- $n > 1$ - Assume $T, \Pi, m \not\models_{\mathsf{hLTL}} \psi_1$ and $\forall o \in \mathbb{N}.\, o \leq n \implies T, \Pi, m-o \not\models_{\mathsf{hLTL}} \psi_2$. From the inner induction hypothesis it follows that $\forall p \in \mathbb{N}.\, p \leq n-1 \implies [\![m-p]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\mathsf{H2M}(\psi_1)\, U\, \mathsf{H2M}(\psi_2)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \emptyset$. It remains to prove the case where $p = n$, that $[\![m-n]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\mathsf{H2M}(\psi_1)\, U\, \mathsf{H2M}(\psi_2)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \emptyset$. Since $T, \Pi, m-n \not\models_{\mathsf{hLTL}} \psi_2$, from the outer induction hypothesis it follows that

  $$[\![m-n]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\mathsf{H2M}(\psi_2)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \emptyset\,.$$

  By (4), it is now enough to prove that

  $$[\![m-n]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap \mathcal{M}^h(T)_\circ(|\mathsf{H2M}(\psi_1)\, U\, \mathsf{H2M}(\psi_2)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}) = \emptyset\,,$$

  which is true if $[\![m-(n-1)]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\mathsf{H2M}(\psi_1)\, U\, \mathsf{H2M}(\psi_2)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \emptyset$, which holds by the (inner) induction hypothesis.

□

Now assume $T, \Pi, i \not\models_{\mathsf{hLTL}} \psi_1\, U\, \psi_2$. By definition of $\models_{\mathsf{hLTL}}$, either there is no $j \geq i$ satisfying $T, \Pi, j \models_{\mathsf{hLTL}} \psi_2$, in which case $\emptyset$ is a fixpoint of F_U, or strictly before first such j there exists some k, $i \leq k < j$ satisfying $T, \Pi, k \not\models_{\mathsf{hLTL}} \psi_1$ (and by choice of j also $T, \Pi, k \not\models_{\mathsf{hLTL}} \psi_2$). Then we can use k and $k-i$ as parameters m and n of the claim above and by the choice $p = k-i$ get $[\![k-(k-i)]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \cap |\mathsf{H2M}(\psi_1)\, U\, \mathsf{H2M}(\psi_2)|_{\rho^{\mathsf{hLTL}}_T(\Pi)} = \emptyset$.

- $\varphi \equiv \exists\pi.\varphi'$ - Assume $T, \Pi, i \vDash_{\mathsf{hLTL}} \exists\pi.\,\psi$. Then there exists some $t \in T$ such that $T, \Pi[\tau/\pi], i \vDash_{\mathsf{hLTL}} \psi$. By the induction hypothesis, $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi)|_{\rho_{\Pi[\tau/\pi]}}$, and since $\rho_{\Pi[\tau/\pi]} = \rho^{\mathsf{hLTL}}_T(\Pi)[(\tau,0)/\pi]$, also $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)[(\tau,0)/\pi]}$. Because $(\tau,0) \in \mathcal{M}^h(T)_{\mathit{Trace}}$, it holds that $\mathcal{M}^h(T) = |\pi \in [\![\mathit{Trace}]\!]|_{\rho^{\mathsf{hLTL}}_T(\Pi)[(\tau,0)/\pi]}$, therefore

$$\begin{aligned}
&[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \\
&\subseteq |(\pi \in [\![\mathit{Trace}]\!]) \wedge \mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)[(\tau,0)/\pi]} \\
&\subseteq \bigcup_{m \in \mathcal{M}^h(T)} |(\pi \in [\![\mathit{Trace}]\!]) \wedge \mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)[m/\pi]} \\
&= |\exists\pi : \mathit{Trace}.\, \mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}\,.
\end{aligned}$$

  For the other implication, assume $(\tau, i) \in |\exists\pi : \mathit{Trace}.\, \mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$ for some $\tau \in T$. Then there must exist some $(\tau', i') \in M_{\mathit{TrSuf}}(T)$ such that $(\tau, i) \in |(\pi \in [\![\mathit{Trace}]\!]) \wedge \mathsf{H2M}(\psi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)[(\tau',i')/\pi]}$. Since

$$(\tau, i) \in |\pi \in [\![\mathit{Trace}]\!]|_{\rho^{\mathsf{hLTL}}_T(\Pi)[(\tau',i')/\pi]}\,,$$

  it follows that $i' = 0$, and therefore $\rho^{\mathsf{hLTL}}_T(\Pi)[(\tau', i')/\pi] = \rho_{\Pi[\tau'/\pi]}$. Then $(\tau, i) \in |\mathsf{H2M}(\psi)|_{\rho_{\Pi[\tau'/\pi]}}$, and from the induction hypothesis, $T, \Pi[\tau'/\pi], i \vDash_{\mathsf{hLTL}} \psi$. But then $T, \Pi, i \vDash_{\mathsf{hLTL}} \exists\pi.\,\psi$.

- $\varphi \equiv \forall\pi.\varphi'$ - Assume $T, \Pi, i \vDash_{\mathsf{hLTL}} \forall\pi.\,\varphi'$. Then for all $\tau \in T$ it holds that $T, \Pi[\tau/\pi], i \vDash_{\mathsf{hLTL}} \varphi'$, and by the induction hypothesis, $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\psi')|_{\rho_{\Pi[\tau/\pi]}} = |\mathsf{H2M}(\psi')|_{\rho^{\mathsf{hLTL}}_T(\Pi)[(\tau,0)/\pi]}$. Since for all $m \notin \mathcal{M}^h(T)_{\mathit{Trace}}$, $|(\pi \in [\![\mathit{Trace}]\!]) \to \varphi'|_{\rho^{\mathsf{hLTL}}_T(\Pi)[m/\pi]} = \mathcal{M}^h(T)$, it follows that

$$\begin{aligned}
[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} &\subseteq \bigcap_{m \in \mathcal{M}^h(T)} |(\pi \in [\![\mathit{Trace}]\!]) \to \varphi'|_{\rho^{\mathsf{hLTL}}_T(\Pi)[m/\pi]} \\
&= |\forall\pi : \mathit{Trace}.\, \mathsf{H2M}(\varphi')|_{\rho^{\mathsf{hLTL}}_T(\Pi)}
\end{aligned}$$

  For the other implication, let $(\tau, i) \in |\forall\pi : \mathit{Trace}.\, \mathsf{H2M}(\varphi')|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$. Therefore for all $(\tau', i') \in \mathcal{M}^h(T)_{\mathit{TrSuf}}$,

$$(\tau, i) \in |(\pi \in [\![\mathit{Trace}]\!]) \to \mathsf{H2M}(\varphi')|_{\rho^{\mathsf{hLTL}}_T(\Pi)[(\tau',i')/\pi]}\,.$$

  Specifically, $(\tau, i) \in |\mathsf{H2M}(\varphi')|_{\rho^{\mathsf{hLTL}}_T(\Pi)[(\tau',0)/\pi]} = |\mathsf{H2M}(\varphi')|_{\rho_{\Pi[\tau'/\pi]}}$ for all $\tau' \in T$, and by the induction hypothesis, $T, \Pi[\tau'/\pi], i \vDash_{\mathsf{hLTL}} \varphi'$. Therefore, $T, \Pi, i \vDash_{\mathsf{hLTL}} \forall\pi.\,\varphi'$.

□

Proof of Theorem 2. $T \vDash_{\mathsf{hLTL}} \varphi$ iff for any valuation $\Pi : V \to T$ holds that $T, \Pi, 1 \vDash_{\mathsf{hLTL}} \varphi$ (by definition), iff for any valuation $\Pi : V \to T$ holds that

$[\![1]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\varphi)|_{\rho^{\mathsf{hLTL}}_T(\Pi)}$ (by Lemma 17), iff for any well-sorted $\mathcal{M}^h(T)$-valuation ρ,

$[\![1]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)} \subseteq |\mathsf{H2M}(\varphi)|_\rho$ (using Lemma 45), iff for any well-sorted $\mathcal{M}^h(T)$-valuation ρ, $\mathcal{M}^h(T) = |[\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)|_\rho$ (since $\mathcal{M}^h(T)_{\mathit{Trace}} = [\![1]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(T)}$), iff for any $\mathcal{M}^h(T)$-valuation ρ, $\mathcal{M}^h(T) = |\chi_{\mathsf{H2M}(\varphi)} \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)|_\rho$, iff $\mathcal{M}^h(T) \vDash \mathsf{WellSorted}(\varphi) \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\varphi)$. □

A.3.2 From Γ^{hLTL}-Models to HyperLTL models

Lemma 49. *Let M be a model of Γ^{hLTL}, and $m \in M_{TrSuf}$. A sequence $m_1, m_2, \dots m_n = m$ where $m_i \in M$ such that $m_1 \in M_{Trace}$ and $m_{i+1} \in M_{\bar{\circ}}(m_i)$ for all $1 \leq i < n$ is called an initial sequence of m. For every m there exists exactly one such sequence; we define a function $\mathit{dist}^{\mathsf{hLTL}}_M \colon M_{TrSuf} \to \mathbb{N}_{\geq 1}$ by $\mathit{dist}^{\mathsf{hLTL}}_M(m) = n$ and a function $\mathit{full}^{\mathsf{hLTL}}_M \colon M_{TrSuf} \to M_{Trace}$ by $\mathit{full}^{\mathsf{hLTL}}_M(m) = m_1$.*

Proof of Lemma 49. For existence, the axiom (TraceSuffix) enforces that $M_{TrSuf} = \mu F$, where $F(A) = M_{Trace} \cup M_{\bar{\circ}}(A)$. Define:

$$\begin{aligned}\xi = \{m \mid & \exists n \in \mathbb{N}_{\geq 1}, \exists m_1, \dots, m_n.\, m_1 = M_{Trace} \wedge m_n = m \\ & \wedge \forall 1 \leq i < n.\, m_{i+1} \in M_{\bar{\circ}}(m_i)\}\end{aligned}$$

and since $F(\xi) \subseteq \xi$, i.e. ξ is a prefix point of F, from the Knaster-Tarski theorem it follows that $M_{TrSuf} \subseteq \xi$, i.e. for every $m \in M_{TrSuf}$ there exists an appropriate sequence.

For uniqueness, let there be two such sequences, $m_1, \dots, m_n$ and $m'_1, \dots, m'_{n'}$, and let l be the length of their maximal common suffix, starting with $m_{n-(l-1)} = m'_{n'-(l-1)}$. Let us assume (w.l.o.g.) that $n \geq n'$. It must be true that $l = n'$, because if $l < n'$, then by the injectivity of $M_{\bar{\circ}}(\_)$ (Lemma 38) there is another common suffix starting with $m_{n-l} = m'_{m'-l}$, which contradicts our choice of maximal common suffix. Since $l = n'$, we have that $m_{n-(l-1)} = m'_{n'-(l-1)} = m'_1 \in M_{Trace}$ But then $M_\circ(m_{n-(l-1)}) = \emptyset$, therefore $m_{n-(l-1)}$ has no predecessors in the sequence $m_1, \dots, m_n$, and $l = n = n'$, and the two sequences are identical. □

Lemma 50. *For every model M, element $m \in M$ and natural number $i \in \mathbb{N}_0$, the following holds:*

1. $\mathit{get}(\mathit{full}^{\mathsf{hLTL}}_M(m), \mathit{dist}^{\mathsf{hLTL}}_M(m)) = m$
2. $\mathit{full}^{\mathsf{hLTL}}_M(\mathit{get}(m, i)) = \mathit{full}^{\mathsf{hLTL}}_M(m)$
3. $\mathit{dist}^{\mathsf{hLTL}}_M(\mathit{get}(m, i)) = i + \mathit{dist}^{\mathsf{hLTL}}_M(m)$

Proof. (1) is proved by induction to the length of the initial sequence of m, (2) and (3) follows from the fact that the initial sequence of $\mathit{get}(m, i)$ is an extension of the initial sequence of m. □

Lemma 51. *For any $M \in \mathbf{Mod}_{\mathsf{ML}}(\Gamma^{\mathsf{hLTL}})$, the function β_M is bijective, and thus its inverse β_M^{-1} exists.*

Proof of Lemma 51. First we prove injectivity. Let $\beta_M(m_1) = \beta_M(m_2)$. Therefore $dist_M^{\mathsf{hLTL}}(m_1) = dist_M^{\mathsf{hLTL}}(m_2)$ and $\tau(full_M^{\mathsf{hLTL}}(m_1)) = \tau(full_M^{\mathsf{hLTL}}(m_2))$. Therefore for all $a \in AP$ and $i \in \mathbb{N}$, $get(full_M^{\mathsf{hLTL}}(m_1), i) \in a_M$ iff $get(full_M^{\mathsf{hLTL}}(m_2), i) \in a_M$, and therefore $M_{eq}(full_M^{\mathsf{hLTL}}(m_1), full_M^{\mathsf{hLTL}}(m_2)) = M$. Since $full_M^{\mathsf{hLTL}}(m_1)$ and $full_M^{\mathsf{hLTL}}(m_2)$ are members of M_{Trace}, from the axiom (SET) it follows that $full_M^{\mathsf{hLTL}}(m_1) = full_M^{\mathsf{hLTL}}(m_2)$. But then from Lemma 50 we have

$$\begin{aligned} m_1 &= get(full_M^{\mathsf{hLTL}}(m_1), dist_M^{\mathsf{hLTL}}(m_1)) \\ &= get(full_M^{\mathsf{hLTL}}(m_2), dist_M^{\mathsf{hLTL}}(m_2)) \\ &= m_2 \,. \end{aligned}$$

For surjectivity, consider some $(\tau, i) \in \mathcal{T}^h(M) \times N$. Then there exists some $m \in M_{Trace}$ such that $\tau = \tau(m)$. Using Lemma 50,

$$\begin{aligned} \beta_M(get(m,i)) &= (\tau(full_M^{\mathsf{hLTL}}(get(m,i))), dist_M^{\mathsf{hLTL}}(get(m,i))) \\ &= (\tau(full_M^{\mathsf{hLTL}}(m)), i + dist_M^{\mathsf{hLTL}}(m)) \\ &= (\tau(m), i) \\ &= (\tau, i) \,. \end{aligned}$$

□

Proof of Lemma **??**.

$$\begin{aligned} \rho_T^{\mathsf{hLTL}}(\Pi(\rho))(\pi) &= (\Pi(\rho)(\pi), 1) \\ &= (\tau(full_M^{\mathsf{hLTL}}(\rho(\pi))), 1) \\ &= (\tau(\rho(\pi)), 1) \\ &= \rho(\pi) \,. \end{aligned}$$

□

Lemma 52. *For any $T \in \mathbf{Mod}_{\mathsf{hLTL}}$, any T-valuation $\Pi : V \to T$, and any $\pi \in V$, $\Pi(\rho_T^{\mathsf{hLTL}}(\Pi))(\pi) = \Pi(\pi)$*

Proof of Lemma 52.

$$\begin{aligned} \Pi(\rho_T^{\mathsf{hLTL}}(\Pi)) &= \Pi(\rho_T^{\mathsf{hLTL}}(\Pi))(\pi) \\ &= \tau(full^{\mathsf{hLTL}}_{\mathcal{M}^h(T)}(\rho_T^{\mathsf{hLTL}}(\Pi)(\pi))) \\ &= \tau(\rho_T^{\mathsf{hLTL}}(\Pi)(\pi)) \\ &= \{(n,a) \mid get(\rho_T^{\mathsf{hLTL}}(\Pi)(\pi), n) \in a_{\mathcal{M}^h(T)}\} \\ &= \{(n,a) \mid (\Pi(\pi), n) \in a_{\mathcal{M}^h(T)}\} \\ &= \{(n,a) \mid a \in \Pi(\pi)(n)\} \\ &= \Pi(\pi) \,. \end{aligned}$$

□

Lemma 53. *$\rho_{\Pi(\rho)}(\pi) = \beta_M(\rho(\pi))$ for all well-sorted valuations ρ and variables $\pi \in V$.*

Proof.

$$\begin{aligned}\rho_{\Pi(\rho)}(\pi) &= (\Pi(\rho)(\pi), 1)\\ &= (\tau(\mathit{full}_M^{\mathsf{hLTL}}(\rho(\pi))), 1)\\ &= (\tau(\mathit{full}_M^{\mathsf{hLTL}}(\rho(\pi))), \mathit{dist}_M^{\mathsf{hLTL}}(\rho(\pi)))\\ &= \beta_M(\rho(\pi))\end{aligned}$$

□

Lemma 54. *For any Γ^{hLTL}-model M, for every trace suffix $m \in M_{TrSuf}$, and every pair $(\tau, l) \in T \times \mathbb{N}_{\geq 1}$,*

1. $\beta_M(M_\circ(m)) = \mathcal{M}^h(\mathcal{T}^h(M))_\circ(\beta_M(m))$,
2. $\beta_M(M_{\bar{\circ}}(m)) = \mathcal{M}^h(\mathcal{T}^h(M))_{\bar{\circ}}(\beta_M(m))$,
3. $M_\circ(\beta_M^{-1}((\tau, l))) = \beta_M^{-1}(\mathcal{M}^h(\mathcal{T}^h(M))_\circ((t, l)))$, *and*
4. $M_{\bar{\circ}}(\beta_M^{-1}(\tau, l)) = \beta_M^{-1}(\mathcal{M}^h(\mathcal{T}^h(M))_{\bar{\circ}}((t, l)))$.

Proof. For (1), if $m \in M_{Trace}$, then by the Lemma 49 there does not exists any $m' \in M$ such that $m \in M_{\bar{\circ}}(m')$, and therefore by the relationship between $M_\circ(\_)$ and $M_{\bar{\circ}}(\_)$ it follows that $M_\circ(m) = \emptyset$, therefore $\beta_M(M_\circ(m)) = \emptyset$. Also, $\beta_M(m) = (\mathit{full}(m), 1)$, therefore $\mathcal{M}^h(\mathcal{T}^h(M))_\circ(\beta_M(m)) = \emptyset$. On the other hand, if $m \notin M_{Trace}$, then from the Lemma 49 and the relationship between $M_\circ(\_)$ and $M_{\bar{\circ}}(\_)$ it follows that $\mathit{dist}(m) > 0$ and that there exists one $m' \in M_\circ(m)$. Then

$$\begin{aligned}\beta_M(M_\circ(m)) &= \beta_M(\{m'\})\\ &= (\mathit{full}(m'), \mathit{dist}(m'))\\ &= (\mathit{full}(m), \mathit{dist}(m) - 1)\\ &= \mathcal{M}^h(\mathcal{T}^h(M))_\circ((\mathit{full}(m), \mathit{dist}(m)))\\ &= \mathcal{M}^h(\mathcal{T}^h(M))_\circ(\beta_M(m))\,.\end{aligned}$$

For (2),

$$\begin{aligned}\beta_M(M_{\bar{\circ}}(m)) &= (\mathit{full}(M_{\bar{\circ}}(m)), \mathit{dist}(M_{\bar{\circ}}(m)))\\ &= (\mathit{full}(m), \mathit{dist}(m) + 1)\\ &= \bar{\circ}_{\mathcal{M}^h(\mathcal{T}^h(M))}((\mathit{full}(m), \mathit{dist}(m)))\\ &= \bar{\circ}_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m))\,.\end{aligned}$$

For (3), it follows from (1) that

$$\begin{aligned}\beta_M(M_\circ(\beta_M^{-1}((\tau, l)))) &= \mathcal{M}^h(\mathcal{T}^h(M))_\circ(\beta_M(\beta_M^{-1}((\tau, l))))\\ &= \mathcal{M}^h(\mathcal{T}^h(M))_\circ((\tau, l))\,,\end{aligned}$$

from which it follows that

$$\begin{aligned} M_\circ(\beta_M^{-1}((\tau,l))) &= \beta_M^{-1}(\beta_M(M_\circ(\beta_M^{-1}((\tau,l))))) \\ &= \beta_M^{-1}(\mathcal{M}^h(\mathcal{T}^h(M))_\circ((\tau,l)))\,. \end{aligned}$$

Point (4) follows from (2) in a similar way. □

Lemma 55. *For any Γ^{hLTL}-model M and any $m \in M_{TrSuf}$ it holds that $row_M(m) = \beta_M^{-1}(row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m)))$.*

Proof. Because $\mathcal{M}^h(\mathcal{T}^h(M))$ satisfies (Row), we may assume that $row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m))$ is the least fixpoint of the function $G_{\beta_M(m)}(X) = \{\beta_M(m)\} \cup \mathcal{M}^h(\mathcal{T}^h(M))_\circ(X) \cup \bar{\circ}_{\mathcal{M}^h(\mathcal{T}^h(M))}(X)$. Since M also satisfies the axiom (Row), we need to show that $\beta_M^{-1}(row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m)))$ is the least fixpoint of the function $F_m(X) = \{m\} \cup \circ MX \cup M_{\bar{\circ}}(X)$. We first show it is a fixpoint:

$$\begin{aligned} &F_m(\beta_M^{-1}(row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m)))) \\ &= \{m\} \cup M_\circ(\beta_M^{-1}(row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m)))) \\ &\cup M_{\bar{\circ}}(\beta_M^{-1}(row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m)))) \\ &= \{m\} \cup \beta_M^{-1}(\mathcal{M}^h(\mathcal{T}^h(M))_\circ(row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m)))) \\ &\cup \beta_M^{-1}(\bar{\circ}_{\mathcal{M}^h(\mathcal{T}^h(M))}(row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m)))) \\ &= \beta_M^{-1}(\{\beta_M(m)\} \cup \circ_{\mathcal{M}^h(\mathcal{T}^h(M))}(row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m))) \\ &\cup \bar{\circ}_{\mathcal{M}^h(\mathcal{T}^h(M))}(row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m)))) \\ &= \beta_M^{-1}(row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m)))\,, \end{aligned}$$

where the second equality holds by Lemma 54 and the last equality by $row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m))$ being a fixpoint of $G_{\beta_M(m)}$. It is also the least fixpoint. Let A be a fixpoint of F_m. Then $\beta_M(A)$ is a fixpoint of $G_{\beta_M(m)}$, since

$$\begin{aligned} G_{\beta_M(m)}(\beta_M(A)) &= \{\beta_M(m)\} \cup \mathcal{M}^h(\mathcal{T}^h(M))_\circ(\beta_M(A)) \\ &\quad \cup \bar{\circ}_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(A)) \\ &= \{\beta_M(m)\} \cup \beta_M(M_\circ(A)) \cup \beta_M(M_{\bar{\circ}}(A)) \\ &= \beta_M(\{m\} \cup M_\circ(A) \cup M_{\bar{\circ}}(A)) \\ &= \beta_M(F_m(A))\,. \\ &= \beta_M(A)\,. \end{aligned}$$

Since $row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m))$ is the least fixpoint of $G_{\beta_M(m)}$, it follows that $row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m)) \subseteq \beta_M(A)$, and therefore $\beta_M^{-1}(row_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m))) \subseteq A$. □

Lemma 56. *For any Γ^{hLTL}-model M and any $m_1, m_2 \in M_{TrSuf}$ it holds that $sc_M(m_1, m_2) = \beta_M^{-1}(sc_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m_1), \beta_M(m_2)))$.*

Proof. For notational simplicity, we write $\langle m_1, m_2, m_3\rangle$ for $(\#\mathsf{pair}, m_1, (\#\mathsf{pair}, m_2, m_3))$, M' for $\mathcal{M}^h(\mathcal{T}^h(M))$, and $X(M, m_1, m_2)$ for $\{m \mid \langle m_1, m_2, m\rangle \in X\}$ whenever X is a set and $m_1, m_2 \in M$. We use the same syntax, $X(M, M_1, M_2)$ to mean the pointwise extension, $\bigcup_{m_1\in M_1, m_2\in M_2} X(M, m_1, m_2)$, whenever $M_1, M_2 \subseteq M$. Since both M and M' satisfy the axiom (SC), it follows that $M_{sc}(m_1, m_2) = (\mu F)(m_1, m_2)$ and ${M'}_{sc}(\beta_M(m_1), \beta_M(m_2)) = (\mu G)(\beta_M(m_1), \beta_M(m_2))$, where

$$\begin{aligned} F(sc) = \bigcup_{m_1,m_2\in M} \{\langle m_1, m_2, \gamma\rangle \mid \gamma \in M \wedge \\ (m_1, m_2 \in M_{\mathit{Trace}} \vee \gamma \in sc(M, M_\circ(m_1), M_\circ(m_2))\}\,, \end{aligned}$$

$$\begin{aligned} G(sc) = \bigcup_{m_1,m_2\in M'} \{\langle m_1, m_2, \gamma)) \mid \gamma \in M' \wedge \\ (m_1, m_2 \in {M'}_{\mathit{Trace}} \vee \gamma \in sc(M, {M'}_\circ(m_1){M'}_\circ(m_2))\}\,. \end{aligned}$$

We will prove that $\mu G = X$, where

$$X = \beta_M(\mu F) = \{(\langle \beta_M(a), \beta_M(b), \beta_M(c)\rangle \mid \langle a, b, c\rangle \in \mu F\}\,,$$

from which it follows that

$$\begin{aligned} &{M'}_{sc}(\beta_M(m_1), \beta_M(m_2)) \\ &= \{\beta_M(c) \mid \langle \beta_M(m_1), \beta_M(m_2), \beta_M(c)\rangle \in \mu G\} \\ &= \{\beta_M(c) \mid \langle m_1, m_2, c\rangle \in \mu F\} \\ &= \beta_M(\{c \mid \langle m_1, m_2, c\rangle \in \mu F\}) \\ &= \beta_M(M_{sc}(m_1, m_2))\,. \end{aligned}$$

First, X is a fixpoint of G:

$$
\begin{aligned}
G(X) = & \bigcup_{p,q \in M'} \{\langle p, q, \gamma\rangle \mid \gamma \in M' \\
& \wedge (p, q \in {M'}_{Trace} \\
& \vee \gamma \in X(M', {M'}_\circ(p), {M'}_\circ(q)))\} \\
= & \bigcup_{p,q \in M} \{\langle \beta_M(p), \beta_M(q), \gamma\rangle \mid \gamma \in M' \\
& \wedge (\beta_M(p), \beta_M(q) \in {M'}_{Trace} \\
& \vee \gamma \in X(M', {M'}_\circ(\beta_M(p)), {M'}_\circ(\beta_M(q))))\} \\
= & \bigcup_{p,q \in M} \{\langle \beta_M(p), \beta_M(q), \gamma\rangle \mid \gamma \in M' \\
& \wedge (p, q \in M_{Trace} \vee \\
& \gamma \in X(M', \beta_M(M_\circ(p)), \beta_M(M_\circ(q))))\} \\
= & \bigcup_{p,q \in M} \{\langle \beta_M(p), \beta_M(q), \gamma\rangle \mid \gamma \in M' \wedge \\
& (p, q \in M_{Trace} \vee \\
& \gamma \in \{d \mid \langle \beta_M(M_\circ(p)), \beta_M(M_\circ(q)), d\rangle \in X\})\} \\
= & \bigcup_{p,q \in M} \langle \beta_M(p), \beta_M(q), r\rangle \mid p, q \in M_{Trace} \\
& \vee \langle \beta_M(M_\circ(p)), \beta_M(M_\circ(q)), r\rangle \in X\} \\
= & \bigcup_{p,q \in M} \{\langle \beta_M(p), \beta_M(q), r\rangle \mid p, q \in M_{Trace} \\
& \vee \langle M_\circ(p), M_\circ(q), \beta_M^{-1}(r)\rangle \in \mu F\} \\
= & \{\langle \beta_M(p), \beta_M(q), r\rangle \mid \langle p, q, \beta_M^{-1}(r)\rangle \in F(\mu F)\} \\
= & \{\langle \beta_M(p), \beta_M(q), \beta_M(r)\rangle \mid \langle p, q, r\rangle \in \mu F\} \\
= & X\,.
\end{aligned}
$$

Now we show that X is the least fixpoint of G. Let A be a fixpoint of G, i.e., $A = G(A)$. We want to show that $X \subseteq A$. It is enough to show that $\mu F = \beta_M^{-1}(X) \subseteq \beta_M^{-1}(A)$, which follows from $\beta_M^{-1}(A)$ being a fixpoint of F, which we prove as follows. We need to show that $F(\beta_M^{-1}(A)) = \beta_M^{-1}(G(A))$. After expansion of F and G and simplification of the second term as in previous part, is remains to be shown that for all $p, q \in M$, $\beta_M^{-1}(A(\beta_M(M_\circ(p)), \beta_M(M_\circ(q)))) =$

$(\beta_M^{-1}(A))(M_\circ(p), M_\circ(q))$, which holds by

$$\begin{aligned}
&\beta_M^{-1}(A(\beta_M(M_\circ(p)), \beta_M(M_\circ(q))))\\
&= \beta_M^{-1}(\{r \mid \langle \beta_M(M_\circ(p)), \beta_M(M_\circ(q)), r\rangle \in A\})\\
&= \{r \mid \langle \beta_M(M_\circ(p)), \beta_M(M_\circ(q)), \beta_M(r)\rangle \in A\}\\
&= \{r \mid \langle M_\circ(p), M_\circ(q), r\rangle \in \beta_M^{-1}(A)\}\\
&= (\beta_M^{-1}(A))(M_\circ(p), M_\circ(q))\,.
\end{aligned}$$

□

Lemma 57. *For any Γ^{hLTL}-model M and any $m \in M_{TrSuf}$ it holds that $col_M(m) = \beta_M^{-1}(col_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m)))$.*

Proof. By the axiom (SC) and Lemma 56,

$$\begin{aligned}
&\beta_M(col_M(m))\\
&= \beta_M(\bigcup_{m' \in M} (\{m'\} \cap sc_M(m, m')))\\
&= \bigcup_{m' \in M} (\{\beta_M(m')\} \cap \beta_M(sc_M(m, m')))\\
&= \bigcup_{m' \in \mathcal{M}^h(\mathcal{T}^h(M))} (\{m'\} \cap \beta_M(sc_M(m, \beta_M^{-1}(m'))))\\
&= \bigcup_{m' \in \mathcal{M}^h(\mathcal{T}^h(M))} (\{m'\} \cap sc_M(\beta_M(m), m')))\\
&= col_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(m))\,.
\end{aligned}$$

□

Lemma 58. *For any Γ^{hLTL}-model M and any $a \in \mathrm{AP}$, $a_{\mathcal{M}^h(\mathcal{T}^h(M))} = \beta_M(a_M)$.*

Proof. From the definitions, we have $\beta_M(a_M) = \{(\tau(\mathit{full}(m)), \mathit{dist}(m)) \mid m \in a_M\}$ and $a_{\mathcal{M}^h(\mathcal{T}^h(M))} = \{(\tau(m), i) \mid m \in M_{Trace} \wedge \mathit{get}(m, i) \in a_M\}$. If $(\tau(m), i) \in a_{\mathcal{M}^h(\mathcal{T}^h(M))}$, we have $m \in M_{Trace}$ and $\mathit{get}(m, i) \in a_M$, and therefore

$$(\tau(\mathit{full}(\mathit{get}(m, i))), \mathit{idx}(\mathit{get}(m, i))) = (\tau(m), i) \in \beta_M(a_M)\,,$$

where the equality holds by Lemma 50. On the other hand, if $(\tau(\mathit{full}(m)), \mathit{dist}(m)) \in \beta_M(a_M)$, we have $\mathit{get}(\mathit{full}(m), \mathit{dist}(m)) = m \in a_M$ and therefore $(\tau(\mathit{full}(m)), \mathit{dist}(m)) \in a_{\mathcal{M}^h(\mathcal{T}^h(M))}$. □

Lemma 59. *$\rho_{\Pi(\rho[m/\pi])}(\pi') = (\rho_{\Pi(\rho)})[\beta_M(m)/\pi](\pi')$ for any model M, well-sorted valuation $\rho : V \to M$, variable $\pi \in V$ and a trace $m \in M_{Trace}$.*

Proof. If $\pi = \pi'$, then

$$\begin{aligned}
\rho_{\Pi(\rho[m/\pi])}(\pi') &= (\Pi(\rho[m/\pi](\pi')), 0) \\
&= (\tau(\mathit{full}(\rho[m/\pi](\pi'))), 0) \\
&= (\tau(m), 0) \\
&= (\tau(\mathit{full}(m), \mathit{dist}(m))) \\
&= \beta_M(m) \\
&= (\rho_{\Pi(\rho)})[\beta_M(m)/\pi](\pi')\,.
\end{aligned}$$

If $\pi \neq \pi'$, then

$$\begin{aligned}
\rho_{\Pi(\rho[m/\pi])}(\pi') &= (\Pi(\rho[m/\pi](\pi')), 0) \\
&= (\tau(\mathit{full}(\rho[m/\pi](\pi'))), 0) \\
&= (\tau(\mathit{full}(\rho(\pi'))), 0) \\
&= (\Pi(\rho)(\pi'), 0) \\
&= \rho_{\Pi(\rho)}(\pi') \\
&= (\rho_{\Pi(\rho)})[\beta_M(m)/\pi](\pi')\,.
\end{aligned}$$

□

Lemma 60. *$\beta_M(\mathit{init}_M) = \mathit{init}_{\mathcal{M}^h(\mathcal{T}^h(M))}$ for all Γ^{hLTL}-models M.*

Proof.

$$\begin{aligned}
\beta_M(\mathit{init}_M) &= \{(\tau(\mathit{full}(m)), \mathit{dist}(m)) \mid m \in M_{\mathit{Trace}}\} \\
&= \{(\tau(m), 0) \mid m \in M_{\mathit{Trace}}\} \\
&= \{(\tau, 0) \mid \tau \in \mathcal{T}^h(M)\} \\
&= \mathcal{M}^h(\mathcal{T}^h(M))_{\mathit{Trace}}
\end{aligned}$$

□

Proof of Lemma 20. By structural induction on φ.

- $\varphi \equiv a_\pi$ - By Lemma 57, Lemma 58, Lemma 55 and Lemma 53,

$$\begin{aligned}
\beta_M(|a_\pi|_\rho) &= \beta_M(|\mathit{col}(a \wedge \mathit{row}(\pi))|_\rho) \\
&= \beta_M(\mathit{col}_M(a_M \cap \mathit{row}_M(\{\rho(\pi)\}))) \\
&= \mathit{col}_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(a_M) \cap \\
&\quad \beta_M(\mathit{row}_M(\{\rho(\pi)\}))) \\
&= \mathit{col}_{\mathcal{M}^h(\mathcal{T}^h(M))}(a_{\mathcal{M}^h(\mathcal{T}^h(M))} \cap \\
&\quad \mathit{row}_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M(\{\rho(\pi)\}))) \\
&= \mathit{col}_{\mathcal{M}^h(\mathcal{T}^h(M))}(a_{\mathcal{M}^h(\mathcal{T}^h(M))} \cap \\
&\quad \mathit{row}_{\mathcal{M}^h(\mathcal{T}^h(M))}(\{\rho_{\Pi(\rho)}(\pi)\})) \\
&= |\mathit{col}(a \wedge \mathit{row}(\pi))|_{\rho_{\Pi(\rho)}} \\
&= |a_\pi|_{\rho_{\Pi(\rho)}}\,.
\end{aligned}$$

- $\varphi \equiv \neg\psi$ - Follows from the induction hypothesis and bijectivity of β_M.
- $\varphi \equiv \psi_1 \vee \psi_2$ - Follows from the induction hypothesis.
- $\varphi \equiv \circ\psi$ -

$$
\begin{aligned}
&|\mathsf{H2M}(\circ\psi)|_{\mathcal{M}^h(\mathcal{T}^h(M))} \\
&= |\circ\mathsf{H2M}(\psi)|_{\mathcal{M}^h(\mathcal{T}^h(M))} \\
&= \mathit{appext}_{\mathcal{M}^h(\mathcal{T}^h(M))}(\{\#\mathsf{next}\}, |\mathsf{H2M}(\psi)|^h_{\mathcal{M}}(\mathcal{T}^h(M))) \\
&= \mathit{appext}_{\mathcal{M}^h(\mathcal{T}^h(M))}(\{\#\mathsf{next}\}, \\
&\beta_M(|\mathsf{WellSorted}(\psi) \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\psi)|_M)) \\
&= \beta_M(M_\circ(|\mathsf{H2M}(\psi)|_M)) \\
&= \beta_M(\mathit{appext}_M(|\circ|_M, |\mathsf{H2M}(\psi)|_M)) \\
&= \beta_M(|\circ\mathsf{H2M}(\psi)|_M) \\
&= \beta_M(|\mathsf{H2M}(\circ\psi)|_M)\,,
\end{aligned}
$$

  where the fourth equality holds for the following reason (recall that translated HyperLTL formulas evaluate to trace suffixes):

$$
\begin{aligned}
&(\tau, n) \in \mathit{appext}_{\mathcal{M}^h(\mathcal{T}^h(M))}(\{\#\mathsf{next}\}, \\
&\beta_M(|\mathsf{WellSorted}(\psi) \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\psi)|_M)) \\
&\quad\iff (\tau, n+1) \in \beta_M(\\
&|\mathsf{WellSorted}(\psi) \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\psi)|_M) \\
&\quad\iff \exists m \in |\mathsf{WellSorted}(\psi) \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\psi)|_M. \\
&(\tau, n+1) = \beta_M(m) \\
&\quad\iff \exists m \in |\mathsf{WellSorted}(\psi) \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\psi)|_M. \\
&\exists m' \in M_\circ(m).\, (\tau, n+1) = \beta_M(m)) \\
&\quad\iff \exists m \in |\mathsf{WellSorted}(\psi) \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\psi)|_M. \\
&\exists m' \in M_\circ(m).\, (\tau, n) = \beta_M(m')) \\
&\quad\iff \exists m' \in M_\circ(|\mathsf{WellSorted}(\psi) \to [\![\mathit{Trace}]\!] \subseteq \mathsf{H2M}(\psi)|_M). \\
&(\tau, n) = \beta_M(m') \\
&\quad\iff (\tau, n) \in \beta_M(M_\circ(|\mathsf{H2M}(\psi)|_M)).
\end{aligned}
$$

- $\varphi \equiv \psi_1\, U\, \psi_2$ - By definition of the U operator, we have

$$
\begin{aligned}
&\beta_M(|\mathsf{H2M}(\psi_1)\, U\, \mathsf{H2M}(\psi_2)|_\rho \\
&= \beta_M(|\mu X.\, \mathsf{H2M}(\psi_2) \vee (\mathsf{H2M}(\psi_1) \wedge \circ X)|_\rho) \\
&= \beta_M(\mu F)\,,
\end{aligned}
$$

  where

$$F(X) = |\mathsf{H2M}(\psi_2)|_\rho \cup (|\mathsf{H2M}(\psi_1)|_\rho \cap M_\circ(X))\,.$$

We also have

$$\begin{aligned}
&|\mathsf{H2M}(\psi_1)\, U\, \mathsf{H2M}(\psi_2)|_{\rho_{\Pi(\rho)}} \\
&= |\mu X.\, \mathsf{H2M}(\psi_2) \vee (\mathsf{H2M}(\psi_1) \wedge \circ X)|_{\rho_{\Pi(\rho)}} \\
&= \mu G\,,
\end{aligned}$$

where

$$\begin{aligned}
G(X) &= |\mathsf{H2M}(\psi_2)|_{\rho_{\Pi(\rho)}} \\
&\quad \cup (|\mathsf{H2M}(\psi_1)|_{\rho_{\Pi(\rho)}} \cap \mathcal{M}^h(\mathcal{T}^h(M))X) \\
&\quad = \beta_M(|\mathsf{H2M}(\psi_2)|_\rho) \\
&\quad \cup (\beta_M(|\mathsf{H2M}(\psi_1)|_\rho) \cap \beta_M(M_\circ(\beta_M^{-1}(X)))) \\
&\quad = \beta_M(|\mathsf{H2M}(\psi_2)|_\rho \\
&\quad \cup (|\mathsf{H2M}(\psi_1)|_\rho \cap M_\circ(\beta_M^{-1}(X))))\,,
\end{aligned}$$

where the second equality follows from the induction hypothesis. Now we need to show that $\beta_M(\mu F) = \mu G$. First, $\beta_M(\mu F)$ is a fixpoint of G:

$$\begin{aligned}
&G(\beta_M(\mu F)) \\
&= \beta_M(|\mathsf{H2M}(\psi_2)|_\rho \cup (|\mathsf{H2M}(\psi_1)|_\rho \cap M_\circ(\mu F))) \\
&= \beta_M(\mu F)\,.
\end{aligned}$$

It is also the least fixpoint. Let A be a fixpoint of G. Then $\beta_M^{-1}(A)$ is a fixpoint of F by

$$\begin{aligned}
&\beta_M^{-1}(A) \\
&= \beta_M^{-1}(G(A)) \\
&= |\mathsf{H2M}(\psi_2)|_\rho \cup (|\mathsf{H2M}(\psi_1)|_\rho \cap M_\circ(\beta_M^{-1}(A))) \\
&= F(\beta_M^{-1}(A))
\end{aligned}$$

and from μF being the least fixpoint of F it follows that $\mu F \subseteq \beta_M^{-1}(A)$, and therefore $\beta_M(\mu F) \subseteq A$.

- $\varphi \equiv \exists \pi.\, \varphi'$ - Using induction hypothesis, Lemma 59, and Lemma 60,

$$\begin{aligned}
&\beta_M(|\exists \pi.\, (\pi \in [\![\mathit{Trace}]\!]) \wedge \mathsf{H2M}(\varphi')|_\rho) \\
&= \bigcup_{m \in M_{\mathit{Trace}}} \beta_M(|\mathsf{H2M}(\varphi')|_{\rho[m/\pi]}) \\
&= \bigcup_{m \in M_{\mathit{Trace}}} |\mathsf{H2M}(\varphi')|_{\rho_{\Pi(\rho[m/\pi])}} \\
&= \bigcup_{m \in M_{\mathit{Trace}}} |\mathsf{H2M}(\varphi')|_{(\rho_{\Pi(\rho)})[\beta_M(m)/\pi]} \\
&= \bigcup_{m \in \mathcal{M}^h(\mathcal{T}^h(M))_{\mathit{Trace}}} |\mathsf{H2M}(\varphi')|_{(\rho_{\Pi(\rho)})[m/\pi]} \\
&= |\exists \pi.\, (\pi \in [\![\mathit{Trace}]\!]) \wedge \mathsf{H2M}(\varphi')|_{\rho_{\Pi(\rho)}}\,.
\end{aligned}$$

Note that $\rho[m/\pi]$ is a well-sorted valuation for any well-sorted valuation ρ and any $m \in M_{Trace}$, for any model M.

- $\varphi \equiv \forall \pi.\, \phi'$ - Similarly to the previous point.

□

Lemma 61. *For every Γ^{hLTL}-model M and every $i \in \mathbb{N}_{\geq 1}$, $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(\mathcal{T}^h(M))} = \beta_M([\![i]\!]^{\mathsf{hLTL}}_M)$*

Proof of Lemma 61. By induction on i.

- $i = 1$ - using Lemma 60,

$$\begin{aligned}[\![1]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(\mathcal{T}^h(M))} &= \mathcal{M}^h(\mathcal{T}^h(M))_{Trace} \\ &= \beta_M(M_{Trace}) \\ &= \beta_M([\![1]\!]^{\mathsf{hLTL}}_M)\,.\end{aligned}$$

- $i > 1$ - by induction hypothesis and Lemma 54,

$$\begin{aligned}[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(\mathcal{T}^h(M))} &= \bar{\circ}_{\mathcal{M}^h(\mathcal{T}^h(M))}([\![i-1]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(\mathcal{T}^h(M))}) \\ &= \bar{\circ}_{\mathcal{M}^h(\mathcal{T}^h(M))}(\beta_M([\![i-1]\!]^{\mathsf{hLTL}}_M)) \\ &= \beta_M(M_{\bar{\circ}}([\![i-1]\!]^{\mathsf{hLTL}}_M)) \\ &= \beta_M([\![i]\!]^{\mathsf{hLTL}}_M)\,.\end{aligned}$$

□

Lemma 62. *For every model M of Γ^{hLTL} and every $i \in \mathbb{N}_{\geq 1}$, $[\![i]\!]^{\mathsf{hLTL}}_M \neq \emptyset$*

Proof. By induction on i using the axiom (Trace). □

Proof of Lemma 21. Since $[\![i]\!]^{\mathsf{hLTL}}_M$ is nonempty (Lemma 62), $[\![i]\!]^{\mathsf{hLTL}}_M \subseteq |\varphi|_\rho$ implies $[\![i]\!]^{\mathsf{hLTL}}_M \cap |\varphi|_\rho \neq \emptyset$. Therefore, it is sufficient to prove just two implications: that $\mathcal{T}^h(M), \Pi(\rho), i \vDash_{\mathsf{hLTL}} \varphi$ implies $[\![i]\!]^{\mathsf{hLTL}}_M \subseteq |\varphi|_\rho$ and that $[\![i]\!]^{\mathsf{hLTL}}_M \cap |\varphi|_\rho \neq \emptyset$ implies $\mathcal{T}^h(M), \Pi(\rho), i \vDash_{\mathsf{hLTL}} \varphi$. For the first implication, assume $\mathcal{T}^h(M), \Pi(\rho), i \vDash_{\mathsf{hLTL}} \varphi$. By Lemma 17, $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(\mathcal{T}^h(M))} \subseteq |\varphi|_{\rho_{\Pi(\rho)}}$, and by Lemma 20 and Lemma 61, $\beta_M([\![i]\!]^{\mathsf{hLTL}}_M) \subseteq \beta_M(|\varphi|_\rho)$. By injectivity of β_M, $[\![i]\!]^{\mathsf{hLTL}}_M \subseteq |\varphi|_\rho$. For the second implication, assume $[\![i]\!]^{\mathsf{hLTL}}_M \cap |\varphi|_\rho \neq \emptyset$. Then $\beta_M([\![i]\!]^{\mathsf{hLTL}}_M) \cap \beta_M(|\varphi|_\rho) \neq \emptyset$, and by Lemma 20 and Lemma 61, $[\![i]\!]^{\mathsf{hLTL}}_{\mathcal{M}^h(\mathcal{T}^h(M))} \cap |\varphi|_{\rho_{\Pi(\rho)}} \neq \emptyset$. By Lemma 17, $\mathcal{T}^h(M), \Pi(\rho), i \vDash_{\mathsf{hLTL}} \varphi$. □

Proof of Theorem 3. $\mathcal{T}^h(M) \vDash_{\mathsf{hLTL}} \varphi$ iff for every valuation $\Pi : V \to \mathcal{T}^h(T)$, $\mathcal{T}^h(T), \Pi, 1 \vDash_{\mathsf{hLTL}} \varphi$, iff for every well-sorted M-valuation ρ, $\mathcal{T}^h(M), \Pi(\rho), 1 \vDash_{\mathsf{hLTL}} \varphi$ (using Lemma 52), iff for every well-sorted M-valuation ρ, $[\![i]\!]^{\mathsf{hLTL}}_M \subseteq |\mathsf{H2M}(\varphi)|_\rho$ (using Lemma 21), iff $M \vDash \mathsf{WellSorted}(\varphi) \to [\![Trace]\!] \subseteq \mathsf{H2M}(\varphi)$ (by the same argument as in the proof of Theorem 2). □

Proof of Theorem 22. For the implication from left to right, let $\models_{\mathsf{LTL}} \varphi$. That means that for any LTL model τ, $\tau \models_{\mathsf{LTL}} \varphi$. But then $\mathcal{T}^l(M) \models_{\mathsf{LTL}} \varphi$, since $\mathcal{T}^l(M)$ is by construction an LTL model. By Theorem 15, for any ML model M such that $M \models \Gamma^{\mathsf{LTL}}$, $M \models_{\mathsf{LTL}} [\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi)$. But then $\Gamma^{\mathsf{LTL}} \models [\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi)$. For the other implication, let $\Gamma^{\mathsf{LTL}} \models [\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi)$. Since for any LTL model τ, $\mathcal{M}^l(\tau) \models \Gamma^{\mathsf{LTL}}$, it follows that $\mathcal{M}^l(\tau) \models [\![\mathit{Trace}]\!] \in \mathsf{L2M}(\varphi)$, and by Theorem 9, $\tau \models_{\mathsf{LTL}} \varphi$. But then $\tau \models_{\mathsf{LTL}} \varphi$. □