# Matching Logic and Modal Logic

**ACM Reference Format:**
. 2018. Matching Logic and Modal Logic. 1, 1 (July 2018), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## TODO LIST



Contents



Author's address:


XXXX-XXXX/2018/7-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

# 1 MATCHING LOGIC PRELIMINARIES

## 1.1 Syntax and Semantics

## 1.2 Sound and Complete Deduction

# 2 IMPORTANT SIGNATURES AND THEORIES

## 2.1 Binders and Fixpoints

Binders can be defined in matching logic. We define two binders $\mu$ and $\nu$ known as the *fixpoint constructs*, where $\mu$ gives the *least fixpoints* (lfp) and $\nu$ gives the *greatest fixpoints* (gfp). Axioms about $\mu$ and $\nu$ are given in the following, where $x$ is a variable, $e$ is a pattern where $x$ does not occur negatively, and $e'$ is a pattern:

| | | | |
|---|---|---|---|
| ($\textsc{Fix}_\mu$) | $\mu x.e = e[\mu x.e/x]$ | ($\textsc{Fix}_\nu$) | $\nu x.e = e[\nu x.e/x]$ |
| ($\textsc{Lfp}$) | $\lfloor e[e'/x] \to e' \rfloor \to (\mu x.e \to e')$ | ($\textsc{Gfp}$) | $\lfloor e' \to e[e'/x] \rfloor \to (e' \to \nu x.e)$ |

The side condition that $x$ does not occur negatively in $e$ guarantees that lfp and gfp exist, by the Knaster-Tarski theorem, so the axiomatization is consistent. Its soundness is also obvious. Notice that the same theorem makes sure that the true lfp and gfp satisfies (Lfp) and (Gfp).

It is impossible to capture the true lfp and gfp in all models, so there are models that interpret $\mu x.e$ and $\nu x.e$ not as the true lfp and gfp. *Intended interpretation* or *intended semantics* means models that interpret $\mu x.e$ and $\nu x.e$ as the true lfp and gfp.

## 2.2 Unlabeled Transition Systems

An *unlabeled transition system* $T = (S, \to)$ consists of a nonempty state set $S$ and a transition relation $\to \subseteq S \times S$. The signature of unlabeled transition systems, denoted as $\Sigma_{\mathrm{UTS}}$, contains a unary symbol $\bullet$ called the "strong-next". Every unlabeled transition system $T = (S, \to)$ can be seen as a $\Sigma_{\mathrm{UTS}}$-model, by taking $S$ as the carrier set and interpreting the unary symbol $\bullet$ as the following function:

$$\bullet_T \colon S \to 2^S \qquad\qquad \bullet_T(b) = \{a \in S \mid a \to b\}$$

In other words, the function $\bullet_T$ gives the set of all *predecessors* with respect to the transition relation. Every $\Sigma_{\mathrm{UTS}}$-model $\mathcal{M} = (M, \_\mathcal{M})$ can be seen as an unlabeled transition system, too, where the state set is $M$ and the transition relation is defined as follows:

$$a \to b \quad \text{iff} \quad a \in \bullet_{\mathcal{M}}(b) \qquad\qquad \text{for every } a, b \in M.$$

Therefore, there is a one-to-one correspondence between $\Sigma_{\mathrm{UTS}}$-models and unlabeled transition systems. From now on, we do not distinguish between the two.

We define the following derived constructs:

$$\begin{aligned} \circ\varphi &\equiv \neg\bullet\neg\varphi && \text{“weak next”} \\ \Diamond\varphi &\equiv \mu f.(\varphi \vee \bullet f) && \text{“eventually”} \\ \Box\varphi &\equiv \nu f.(\varphi \wedge \circ f) && \text{“always”} \\ \varphi_1 \mathbin{\mathsf{U}_s} \varphi_2 &\equiv \mu f.(\varphi_2 \vee (\varphi_1 \wedge \bullet f)) && \text{“strong until”} \end{aligned}$$

Add a "weak until" here, following the U in finite-trace LTL.

whose intended semantics (Table 1), as we will see, is very well suggested by their names.

The *theory of unlabeled transition systems*, denoted as UTS, is a theory of signature $\Sigma_{\mathrm{UTS}}$ with no axioms, so all unlabeled transition systems are models of the theory UTS. If we want to consider

| Let $\varphi$ be a pattern, $T = (S, \to)$ be an unlabeled transition system, $a$ be a state, and $\rho$ be a valuation. | | |
|---|---|---|
| $a \in \bar{\rho}(\bullet\varphi)$ | iff there exists $b \in S$ such that $a \to b$ and $b \in \bar{\rho}(\varphi)$ | "strong next" |
| $a \in \bar{\rho}(\circ\varphi)$ | iff for all $b \in S$, if $a \to b$ then $b \in \bar{\rho}(\varphi)$ | "weak next" |
| $a \in \bar{\rho}(\Diamond\varphi)$ | iff there exists $n \geq 0$ and $b_1, \ldots, b_n \in S$ such that $a \to b_1$, $\ldots$, $b_{n-1} \to b_n$, and $b_n \in \bar{\rho}(\varphi)$ | "eventually" |
| $a \in \bar{\rho}(\Box\varphi)$ | iff there for all $n \geq 0$ and $b_1, \ldots, b_n \in S$, if $a \to b_1$, $\ldots$, $b_{n-1} \to b_n$ then $b_n \in \bar{\rho}(\varphi)$ | "always" |
| $a \in \bar{\rho}(\varphi_1 \mathsf{U}_s \varphi_2)$ | iff there exists $n \geq 0$ and $b_1, \ldots, b_n \in S$ such that $a \to b_1$, $\ldots$, $b_{n-1} \to b_n$, and $b_n \in \bar{\rho}(\varphi_2)$, and $a, b_1, \ldots, b_{n-1} \in \varphi_1$ | "strong until" |

Table 1. Intended semantics of derived constructs in unlabeled transition systems

only certain types of unlabeled transition systems, we can add additional axioms to UTS. Here, we introduce two important types of them.

*Linear unlabeled transition systems* are unlabeled transition systems $T = (S, \to)$ where the relation $\to$ is a *partial function*:

$$a \to b \text{ and } a \to c \text{ imply } b = c \qquad \text{for every } a, b, c \in S.$$

The *theory of linear unlabeled systems* adds the following additional axiom to the theory UTS:

$$(\textsc{Lin}) \qquad \bullet\varphi \to \circ\varphi$$

Finite and infinite unlabeled transition systems.

$$(\textsc{Inf}) \quad \bullet\top \qquad\qquad (\textsc{Fin}) \quad \Diamond\circ\bot$$

### 2.3 Labeled Transition Systems

A *labeled transition system* $T = (S, A, \{\xrightarrow{a}\}_{a \in A})$ consists of a nonempty state set $S$, a nonempty action set $A$, and an $A$-indexed set of transition relations.

## 3 MODAL LOGIC VARIANTS AS MATCHING LOGIC THEORIES

### 3.1 Linear Temporal Logic

Linear temporal logic (LTL) is parametric on a countably infinite nonempty set AP of *atomic propositions*, which are denoted by $p, q, r$, etc. In literature, the term LTL often refers to the *infinite-trace LTL*, where LTL formulas are interpreted on *infinite traces*. In program verification and especially runtime verification [? ], finite execution traces play an important role, and thus *finite-trace LTL* is considered. In this section, we will show how both finite- and infinite-trace LTL are instances of matching logic.

*3.1.1 Infinite-trace LTL.* Readers should be more familiar with infinite-trace LTL, so let us consider that. The *syntax* of infinite-trace LTL extends the syntax of propositional calculus with a "next" modality $\circ$ and a "strong until" modality $\mathsf{U}_s$:

$$\varphi \quad ::= \quad p \in \mathrm{AP} \mid \varphi \wedge \varphi \mid \neg\varphi \mid \circ\varphi \mid \varphi \mathsf{U}_s \varphi$$

Other proposition connectives can be defined in the usual way. In particular, we use *true* and *false* to denote the propositional true and false. Common temporal modalities can be introduced in the usual way as follows.

$$\Diamond\varphi \equiv \mathit{true} \mathsf{U}_s \varphi \qquad \text{"eventually"}$$

$$\Box\varphi \equiv \neg(\Diamond\neg\varphi) \qquad \text{"always"}$$

*proof system of propositional calculus extended with the following:*

| | |
|---|---|
| ($K_\bigcirc$) | $\bigcirc(\varphi_1 \to \varphi_2) \to (\bigcirc\varphi_1 \to \bigcirc\varphi_2)$ |
| ($N_\bigcirc$) | $\dfrac{\varphi}{\bigcirc\varphi}$ |
| ($K_\Box$) | $\Box(\varphi_1 \to \varphi_2) \to (\Box\varphi_1 \to \Box\varphi_2)$ |
| ($N_\Box$) | $\dfrac{\varphi}{\Box\varphi}$ |
| (Fun) | $\bigcirc\varphi \leftrightarrow \neg(\bigcirc\neg\varphi)$ |
| ($U_1$) | $(\varphi_1 \mathsf{U}_s \varphi_2) \to \Diamond\varphi_2$ |
| ($U_2$) | $(\varphi_1 \mathsf{U}_s \varphi_2) \leftrightarrow (\varphi_2 \vee (\varphi_1 \wedge \bigcirc(\varphi_1 \mathsf{U}_s \varphi_2)))$ |
| (Ind) | $\Box(\varphi \to \bigcirc\varphi) \to (\varphi \to \Box\varphi)$ |

Fig. 1. A sound and complete proof system for infinite-trace LTL

Infinite-trace LTL formulas are interpreted on *infinite traces of sets of atomic propositions*, denoted as $\textsc{Traces}^\omega = [\mathbb{N} \to 2^{\text{AP}}]$. We use $\alpha = \alpha_0\alpha_1 \ldots$ to denote an infinite trace, and use the conventional notation $\alpha_{\geq i}$ to denote the suffix trace $\alpha_i\alpha_{i+1} \ldots$. We define an LTL formula $\varphi$ holds on an infinite trace $\alpha$, denoted as $\alpha \vDash_{\text{infLTL}} \varphi$, in the following inductive way:

- $\alpha \vDash_{\text{infLTL}} p$ if $p \in \alpha_0$ for atomic proposition $p$;
- $\alpha \vDash_{\text{infLTL}} \varphi_1 \wedge \varphi_2$ if $\alpha \vDash_{\text{infLTL}} \varphi_1$ and $\alpha \vDash \varphi_2$;
- $\alpha \vDash_{\text{infLTL}} \neg\varphi$ if $\alpha \nvDash_{\text{infLTL}} \varphi$;
- $\alpha \vDash_{\text{infLTL}} \bigcirc\varphi$ if $\alpha_{\geq 1} \vDash_{\text{infLTL}} \varphi$;
- $\alpha \vDash_{\text{infLTL}} \varphi_1 \mathsf{U}_s \varphi_2$ if there is $j \geq 0$ such that $\alpha_{\geq j} \vDash_{\text{infLTL}} \varphi_2$ and for every $i < j$, $\alpha_{\geq i} \vDash_{\text{infLTL}} \varphi_1$.

An infinite-trace LTL formula $\varphi$ is *valid*, denoted as $\vDash_{\text{infLTL}} \varphi$, if it holds on every infinite trace, i.e., $\alpha \vDash_{\text{infLTL}} \varphi$ for every $\alpha \in \textsc{Traces}^\omega$. A proof system for infinite-trace LTL is shown in Figure 1. We write $\vdash_{\text{infLTL}} \varphi$ if $\varphi$ can be derived using the proof system. A soundness and completeness result is established for infinite-trace LTL [? ], stated as follows.

Theorem 3.1 (Soundness and Completeness for Infinite-trace LTL). *If $\varphi$ is an infinite-trace LTL formula, then $\vDash_{\text{infLTL}} \varphi$ if and only if $\vdash_{\text{infLTL}} \varphi$.*

We next show that we can define a matching logic theory $\mathsf{LTL}_{(\textsc{Inf})}$ which faithfully captures infinite-trace LTL. The theory $\mathsf{LTL}_{(\textsc{Inf})}$ extends the theory UTS with a constant symbol $p$ for every atomic proposition $p \in \text{AP}$. Thanks to the derived constructs we define in Section 2.2,

*Any infinite-trace LTL formula is a pattern in the theory* $\mathsf{LTL}_{(\textsc{Inf})}$.

The theory $\mathsf{LTL}_{(\textsc{Inf})}$ has two axioms:

$$(\textsc{Lin})\quad \bullet\varphi \to \bigcirc\varphi \qquad\qquad (\textsc{Inf})\quad \bullet\top$$

The *standard model* of the theory $\mathsf{LTL}_{(\textsc{Inf})}$ is an unlabeled transition system $(\textsc{Traces}^\omega, \to)$ where the state set is the set of all infinite traces, and the transition relation is defined as follows:

$$\alpha \to \beta \qquad \text{if and only if} \qquad \beta = \alpha_{\geq 1}.$$

The standard model interprets every constant symbol $p$ to the set of all traces $\alpha$ such that $p \in \alpha_0$. In addition, fixpoint constructs $\mu$ and $\nu$ are interpreted as the true lfp and gfp in the model. In the following, whenever we say "standard models", we implicitly mean that the fixpoint constructs are interpreted by their intended semantics. By abuse of notation, we refer to this standard model as also $\textsc{Traces}^\omega$. It is easy to verify that $\textsc{Traces}^\omega$ satisfies (Lin) and (Inf).

Proposition 3.2. *If $\varphi$ is an infinite-trace LTL formula, then $\vDash_{\text{infLTL}} \varphi$ if and only if* $\textsc{Traces}^{\omega} \vDash \varphi$.

Proof. Let $\rho$ be any valuation. Since $\varphi$ is an LTL formula, it contains no variables and thus it does not matter which valuation we choose. We will prove that $\alpha \vDash_{\text{infLTL}} \varphi$ if and only if $\alpha \in \bar{\rho}(\varphi)$, for every $\alpha \in \textsc{Traces}^{\omega}$, by structural induction on the formula $\varphi$. If $\varphi$ is an atomic proposition $p$,

$$\alpha \vDash_{\text{infLTL}} p \quad \text{if and only if} \quad p \in \alpha_0 \quad \text{if and only if} \quad \alpha \in \bar{\rho}(p).$$

If $\varphi \equiv \varphi_1 \wedge \varphi_2$, $\alpha \vDash_{\text{infLTL}} \varphi_1 \wedge \varphi_2$ if and only if $\alpha \vDash_{\text{infLTL}} \varphi_1$ and $\alpha \vDash_{\text{infLTL}} \varphi_2$. By induction hypothesis, $\alpha \in \bar{\rho}(\varphi_1)$ and $\alpha \in \bar{\rho}(\varphi_2)$, and thus $\alpha \in \bar{\rho}(\varphi_1 \wedge \varphi_2)$. If $\varphi \equiv \neg\varphi_1$, $\alpha \vDash_{\text{infLTL}} \neg\varphi_1$ if and only if $\alpha \nvDash_{\text{infLTL}} \varphi_1$.

Finish this proof.

□

Notice that both the proof system for infinite-trace LTL and the one for matching logic are complete. As a direct corollary of Proposition 3.2, the following result holds.

Theorem 3.3 (Conservative Extension for Infinite-Trace LTL). *If $\varphi$ is an infinite-trace LTL formula, then $\vdash_{\text{infLTL}} \varphi$ if and only if* $\mathsf{LTL}_{(\textsc{Inf})} \vdash \varphi$.

Proof. ($\Longleftarrow$). Assume $\nvdash_{\text{infLTL}} \varphi$. By completeness, $\nvDash_{\text{infLTL}} \varphi$, and thus there exists an infinite trace $\alpha$ such that $\alpha \nvDash_{\text{infLTL}} \varphi$. By Proposition 3.2, $\textsc{Traces}^{\omega} \nvDash \varphi$. Because $\textsc{Traces}^{\omega} \vDash \mathsf{LTL}_{(\textsc{Inf})}$, we have $\mathsf{LTL}_{(\textsc{Inf})} \nvDash \varphi$, and by completeness, $\mathsf{LTL}_{(\textsc{Inf})} \nvdash \varphi$.

($\Longrightarrow$). It suffices to prove that all axioms and rules in Figure 1 are provable in matching logic. □

*3.1.2 Finite-trace LTL.* Unlike infinite-trace LTL, finite-trace LTL formulas are interpreted on finite traces. The *syntax* of finite-trace LTL extends the syntax of propositional calculus with a unary "weak next" $\circ$, and a binary "until" $\mathsf{U}$. Derived constructs can be defined in their usual way, too.

$$\varphi \quad ::= \quad p \in \text{AP} \mid \varphi \wedge \varphi \mid \neg\varphi \mid \circ\varphi \mid \varphi \,\mathsf{U}\, \varphi$$

$$\begin{aligned} \Box\varphi &\equiv \varphi \,\mathsf{U}\, \mathit{false} && \text{"always"} \\ \Diamond\varphi &\equiv \neg(\Box\neg\varphi) && \text{"eventually"} \\ \varphi_1 \,\mathsf{U}_s\, \varphi_2 &\equiv \Diamond\varphi_2 \wedge (\varphi_1 \,\mathsf{U}\, \varphi_2) && \text{"strong until"} \end{aligned}$$

Finite-trace LTL formulas are interpreted on nonempty finite traces of sets of atomic propositions, denoted as $\alpha = \alpha_0 \ldots \alpha_n$ and $n \geq 0$ is called the *length* of the trace. The semantics $\alpha \vDash_{\text{finLTL}} \varphi$ is defined inductively on the structure of $\varphi$ as follows:

- $\alpha_0 \ldots \alpha_n \vDash_{\text{finLTL}} p$ if $p \in \alpha_0$ for atomic proposition $p$;
- $\alpha_0 \ldots \alpha_n \vDash_{\text{finLTL}} \varphi_1 \wedge \varphi_2$ if $\alpha_0 \ldots \alpha_n \vDash_{\text{finLTL}} \varphi_1$ and $\alpha_0 \ldots \alpha_n \vDash \varphi_2$;
- $\alpha_0 \ldots \alpha_n \vDash_{\text{finLTL}} \neg\varphi$ if $\alpha_0 \ldots \alpha_n \nvDash_{\text{finLTL}} \varphi$;
- $\alpha_0 \ldots \alpha_n \vDash_{\text{finLTL}} \circ\varphi$ if $n = 0$ or $\alpha_1 \ldots \alpha_n \vDash_{\text{finLTL}} \varphi$;
- $\alpha_0 \ldots \alpha_n \vDash_{\text{finLTL}} \varphi_1 \,\mathsf{U}\, \varphi_2$ if either for every $i \leq n$, $s_i \ldots s_n \vDash_{\text{finLTL}} \varphi_1$, or there is $j \leq n$ such that $\alpha_j \ldots \alpha_n \vDash_{\text{finLTL}} \varphi_2$ and for every $i < j$, $\alpha_i \ldots \alpha_n \vDash_{\text{finLTL}} \varphi_1$.

Finite-trace LTL has a sound and complete proof system as shown in Figure 2.

We can define a matching logic theory $\mathsf{LTL}_{(\textsc{Fin})}$, extending the theory UTS, which captures faithfully finite-trace LTL. The theory $\mathsf{LTL}_{(\textsc{Fin})}$ defines a constant symbol $p$ for every atomic proposition $p \in \text{AP}$. Thanks to the derived constructs we define in Section 2.2,

*Any finite-trace LTL formula is a pattern in the theory* $\mathsf{LTL}_{(\textsc{Fin})}$.

*proof system of propositional calculus extended with the following:*

| | |
|---|---|
| $(\mathrm{K}_{\bigcirc})$ | $\bigcirc(\varphi_1 \to \varphi_2) \to (\bigcirc\varphi_1 \to \bigcirc\varphi_2)$ |
| $(\mathrm{N}_{\bigcirc})$ | $\dfrac{\varphi}{\bigcirc\varphi}$ |
| $(\mathrm{K}_{\Box})$ | $\Box(\varphi_1 \to \varphi_2) \to (\Box\varphi_1 \to \Box\varphi_2)$ |
| $(\mathrm{N}_{\Box})$ | $\dfrac{\varphi}{\Box\varphi}$ |
| $(\neg\bigcirc)$ | $\bigcirc\varphi \leftrightarrow \neg(\bigcirc\neg\varphi)$ |
| (Fix) | $(\varphi_1 \mathbin{\mathsf{U}} \varphi_2) \leftrightarrow (\varphi_2 \vee (\varphi_1 \wedge \bigcirc(\varphi_1 \mathbin{\mathsf{U}} \varphi_2)))$ |
| (coInd) | $\dfrac{\bigcirc\varphi \to \varphi}{\varphi}$ |

Fig. 2. A sound and complete proof system for finite-trace LTL

The theory $\mathsf{LTL}_{(\textsc{Fin})}$ has two axioms:

$$(\textsc{Lin}) \quad \bullet\varphi \to \bigcirc\varphi \qquad\qquad (\textsc{Fin}) \quad \Diamond\bigcirc\bot$$

The *standard model* for the theory $\mathsf{LTL}_{(\textsc{Fin})}$ is an unlabeled transition system $(\textsc{Traces}^*, \to)$ where the state set is the set of all nonempty finite traces, and the transition relation is defined as:

$$\alpha_0 \dots \alpha_n \to \alpha_1 \dots \alpha_n \quad \text{if } n > 0.$$

In addition, every constant symbol $p$ is interpreted to the set of all traces $\alpha_0 \dots \alpha_n$ such that $p \in \alpha_0$. By abuse of notation, we refer to this standard model as also $\textsc{Traces}^*$, and it is easy to prove that $\textsc{Traces}^*$ satisfies (Lin) and (Fin).

Proposition 3.4. *If $\varphi$ is a finite-trace LTL formula, then $\vDash_{\mathrm{finLTL}} \varphi$ if and only if $\textsc{Traces}^* \vDash \varphi$.*

Proof. We will prove the following stronger result. If $\varphi$ is a finite-trace LTL formula and $\alpha \in \textsc{Traces}^*$ is a finite trace, then $\alpha \vDash_{\mathrm{finLTL}} \varphi$ if and only if $\alpha \in \bar{\rho}(\varphi)$, where $\rho$ is any valuation. □

Theorem 3.5 (Conservative Extension for Finite-Trace LTL). *If $\varphi$ is an finite-trace LTL formula, then $\vdash_{\mathrm{finLTL}} \varphi$ if and only if $\mathsf{LTL}_{(\textsc{Fin})} \vdash \varphi$.*

### 3.2 Computation Tree Logic

Computation tree logic (CTL) is a basic and popular *branching-time* temporal logic. The *syntax* is parametric on a nonempty set of atomic propositions AP, and extends propositional calculus with two unary modalities AX "all-path next" and EX "one-path next", and two binary modalities AU "all-path until" and EU "one-path until". Derived constructs can be defined in the usual way.

$$\varphi \quad ::= \quad p \in \mathsf{AP} \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathsf{AX}\varphi \mid \mathsf{EX}\varphi \mid \varphi \mathbin{\mathsf{AU}} \varphi \mid \varphi \mathbin{\mathsf{EU}} \varphi$$

$$\begin{aligned} \mathsf{EF}\varphi &\equiv \mathit{true} \mathbin{\mathsf{EU}} \varphi && \text{"one-path eventually"} \\ \mathsf{AG}\varphi &\equiv \neg(\mathsf{EF}\neg\varphi) && \text{"all-path always"} \\ \mathsf{AF}\varphi &\equiv \mathit{true} \mathbin{\mathsf{AU}} \varphi && \text{"all-path eventually"} \\ \mathsf{EG}\varphi &\equiv \neg(\mathsf{AF}\neg\varphi) && \text{"one-path always"} \end{aligned}$$

OK. I know CTL has many semantics. The following one is somehow nonstandard, but convenient. We need a citation here.

CTL formulas are interpreted on nonempty *infinite trees*, whose nodes are sets of atomic propositions. The set of all infinite trees is denoted as $\textsc{Trees}^\omega$. Here, a tree is *infinite* in the sense that it has no leaves. We use $\tau$ to denote a tree, $\mathrm{root}(\tau)$ to denote its root (which is a set of atomic propositions) and $\mathrm{subtrees}(\tau)$ to denote the set of all subtrees. Since $\tau$ is infinite, $\mathrm{subtrees}(\tau)$ is never empty. The semantics $\tau \vDash_{\mathrm{CTL}} \varphi$ is defined as the *smallest* relation that satisfies the following:

- $\tau \vDash_{\mathrm{CTL}} p$ if $p \in \mathrm{root}(\tau)$ for atomic proposition $p$;
- $\tau \vDash_{\mathrm{CTL}} \varphi_1 \wedge \varphi_2$ if $\tau \vDash_{\mathrm{CTL}} \varphi_1$ and $\tau \vDash \varphi_2$;
- $\tau \vDash_{\mathrm{CTL}} \neg\varphi$ if $\tau \nvDash_{\mathrm{CTL}} \varphi$;
- $\tau \vDash_{\mathrm{CTL}} \mathsf{AX}\varphi$ if for every $\tau' \in \mathrm{subtrees}(\tau)$, $\tau' \vDash_{\mathrm{CTL}} \varphi$;
- $\tau \vDash_{\mathrm{CTL}} \mathsf{EX}\varphi$ if there exists $\tau' \in \mathrm{subtrees}(\tau)$, $\tau' \vDash_{\mathrm{CTL}} \varphi$;
- $\tau \vDash_{\mathrm{CTL}} \varphi_1 \mathsf{AU} \varphi_2$ if either $\tau \vDash_{\mathrm{CTL}} \varphi_2$, or $\tau \vDash_{\mathrm{CTL}} \varphi_1$ and for every $\tau' \in \mathrm{subtrees}(\tau)$, $\tau' \vDash_{\mathrm{CTL}} \varphi_1 \mathsf{AU} \varphi_2$
- $\tau \vDash_{\mathrm{CTL}} \varphi_1 \mathsf{EU} \varphi_2$ if either $\tau \vDash_{\mathrm{CTL}} \varphi_2$, or $\tau \vDash_{\mathrm{CTL}} \varphi_1$ and there exists $\tau' \in \mathrm{subtrees}(\tau)$, $\tau' \vDash_{\mathrm{CTL}} \varphi_1 \mathsf{EU} \varphi_2$

CTL has a sound and complete proof system as shown in Figure 3.

We can define a matching logic theory CTL that faithfully captures CTL. The theory CTL extends the theory UTS by defining a constant symbol $p$ for every atomic proposition $p \in \mathrm{AP}$. We also define the syntax of CTL as derived matching logic constructs as follows:

$$\mathsf{AX}\varphi \equiv \circ\varphi \qquad \varphi_1 \mathsf{AU} \varphi_2 \equiv \mu f.\varphi_2 \vee (\varphi_1 \wedge \circ f)$$
$$\mathsf{EX}\varphi \equiv \bullet\varphi \qquad \varphi_1 \mathsf{EU} \varphi_2 \equiv \mu f.\varphi_2 \vee (\varphi_1 \wedge \bullet f)$$

With these derived constructs,

*Any CTL formula is a pattern of the theory* CTL.

The theory CTL has only one axiom

$$(\textsc{Inf}) \quad \bullet\top.$$

The *standard model* of the theory CTL is an unlabeled transition system $(\textsc{Trees}^\omega, \to)$ where the state set is the set of all infinite trees and the transition relation is defined as

$$\tau \to \tau' \quad \text{if and only if} \quad \tau' \in \mathrm{subtrees}(\tau)$$

In addition, every constant symbol $p$ is interpreted to the set of all infinite trees $\tau$ such that $p \in \mathrm{root}(\tau)$. By abuse of notation, we refer to this standard model as also $\textsc{Trees}^\omega$. It is easy to prove that $\textsc{Trees}^\omega$ satisfies (Inf).

Proposition 3.6. *Let $\rho$ be any valuation. If $\varphi$ is a CTL formula and $\tau$ is an infinite tree, then $\tau \vDash_{\mathrm{CTL}} \varphi$ if and only if $\tau \in \bar{\rho}(\varphi)$. As a corollary, $\vDash_{\mathrm{CTL}} \varphi$ if and only if $\textsc{Trees}^\omega \vDash \varphi$.*

Proof. Since $\varphi$ is a CTL formula, it contains no variables (as a matching logic pattern), and thus the valuation $\rho$ does not really matter. The proof is by structural induction on the formula $\varphi$, but we will prove the "if" part ($\Longleftarrow$) and the "only if" part ($\Longrightarrow$) separately. Recall that we define the CTL semantics $\tau \vDash_{\mathrm{CTL}} \varphi$ as the *smallest* relation that satisfies certain conditions, so to prove ($\Longrightarrow$), it suffices to show $\tau \in \bar{\rho}(\varphi)$ also satisfies the same conditions. The proof of ($\Longleftarrow$) is by the fact that we define AU and EU as least fixpoints in matching logic, and the standard model $\textsc{Trees}^\omega$ adopts the intended semantics.

($\Longrightarrow$). We verify that $\tau \in \bar{\rho}(\varphi)$, as a binary relation between infinite trees and CTL formulas, satisfies the same set of conditions as the CTL semantics $\tau \vDash_{\mathrm{CTL}} \varphi$. The proof is by carrying out a case analysis. If $\varphi \equiv p$ is an atomic proposition, then $p \in \mathrm{root}(\tau)$ if and only if $\tau \in \bar{\rho}(p)$ by the construction of the standard model. The cases where $\varphi \equiv \neg\varphi_1$ and $\varphi \equiv \varphi_1 \wedge \varphi_2$ are trivial. For the

*proof system of propositional calculus extended with the following:*

| | |
|---|---|
| (CTL1) | $\mathsf{EX}(\varphi_1 \vee \varphi_2) \leftrightarrow \mathsf{EX}\varphi_1 \vee \mathsf{EX}\varphi_2$ |
| (CTL2) | $\mathsf{AX}\varphi \leftrightarrow \neg(\mathsf{EX}\neg\varphi)$ |
| (CTL3) | $\varphi_1\ \mathsf{EU}\ \varphi_2 \leftrightarrow \varphi_2 \vee (\varphi_1 \wedge \mathsf{EX}(\varphi_1\ \mathsf{EU}\ \varphi_2))$ |
| (CTL4) | $\varphi_1\ \mathsf{AU}\ \varphi_2 \leftrightarrow \varphi_2 \vee (\varphi_1 \wedge \mathsf{AX}(\varphi_1\ \mathsf{AU}\ \varphi_2))$ |
| (CTL5) | $\mathsf{EX}\mathit{true} \wedge \mathsf{AX}\mathit{true}$ |
| (CTL6) | $\mathsf{AG}(\varphi_3 \rightarrow (\neg\varphi_2 \wedge \mathsf{EX}\varphi_3)) \rightarrow (\varphi_3 \rightarrow \neg(\varphi_1\ \mathsf{AU}\ \varphi_2))$ |
| (CTL7) | $\mathsf{AG}(\varphi_3 \rightarrow (\neg\varphi_2 \wedge (\varphi_1 \rightarrow \mathsf{AX}\varphi_3))) \rightarrow (\varphi_3 \rightarrow \neg(\varphi_1\ \mathsf{EU}\ \varphi_2))$ |
| (CTL8) | $\mathsf{AG}(\varphi_1 \rightarrow \varphi_2) \rightarrow (\mathsf{EX}\varphi_1 \rightarrow \mathsf{EX}\varphi_2)$ |

Fig. 3. A sound and complete proof system for CTL

case where $\varphi \equiv \mathsf{AX}\varphi_1$, if for every $\tau' \in \text{subtrees}(\tau)$, $\tau' \in \bar{\rho}(\varphi_1)$,

$$\begin{array}{ll} \text{then} & \text{for every } \tau' \text{ such that } \tau \rightarrow \tau',\ \tau' \in \bar{\rho}(\varphi_1) \\ \text{then} & \tau \in \bar{\rho}(\bigcirc\varphi_1) \\ \text{then} & \tau \in \bar{\rho}(\mathsf{AX}\varphi_1). \end{array}$$

The case where $\varphi \equiv \mathsf{EX}\varphi_1$ is similar. For the case where $\varphi \equiv \varphi_1\ \mathsf{AU}\ \varphi_2$, if $\tau \in \bar{\rho}(\varphi_2)$, then by definition of AU, $\tau \in \bar{\rho}(\varphi_1\ \mathsf{AU}\ \varphi_2)$. If $\tau \in \bar{\rho}(\varphi_1)$ and for every $\tau' \in \text{subtrees}(\tau)$, $\tau' \in \bar{\rho}(\varphi_1\ \mathsf{AU}\ \varphi_2)$,

$$\begin{array}{ll} \text{then} & \tau \in \bar{\rho}(\varphi_1) \text{ and for every } \tau' \text{ such that } \tau \rightarrow \tau',\ \tau' \in \bar{\rho}(\varphi_1\ \mathsf{AU}\ \varphi_2) \\ \text{then} & \tau \in \bar{\rho}(\varphi_1) \text{ and } \tau \in \bar{\rho}(\bigcirc(\varphi_1\ \mathsf{AU}\ \varphi_2)) \\ \text{then} & \tau \in \bar{\rho}(\varphi_1 \wedge \bigcirc(\varphi_1\ \mathsf{AU}\ \varphi_2))\ , \end{array}$$

and by definition of AU, $\tau \in \bar{\rho}(\varphi_1\ \mathsf{AU}\ \varphi_2)$. The case for $\varphi \equiv \varphi_1\ \mathsf{EU}\ \varphi_2$ is similar. We conclude that $\tau \in \bar{\rho}(\varphi)$ satisfies the same set of conditions that the CTL semantics $\tau \models_{\text{CTL}} \varphi$ satisfies. Since $\tau \models_{\text{CTL}} \varphi$ is defined as the *smallest* relation, we know $\tau \models_{\text{CTL}} \varphi$ implies $\tau \in \bar{\rho}(\varphi)$, and here ends the proof of ($\Longrightarrow$).

□