Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
Electronic Notes in Theoretical Computer Science 262 (2010) 189–204
www.elsevier.com/locate/entcs

A History of Until

Andrea Masini Luca Viganò Marco Volpe

Department of Computer Science, University of Verona, Italy
Email: {andrea.masini | luca.vigano | marco.volpe}@univr.it

Abstract

Until is a notoriously difficult temporal operator as it is both existential and universal at the same time: $A\mathsf{U}B$ holds at the current time instant w iff either B holds at w or there *exists* a time instant w' in the future at which B holds and such that A holds in *all* the time instants between the current one and w'. This "ambivalent" nature poses a significant challenge when attempting to give deduction rules for until. In this paper, in contrast, we make explicit this duality of until by introducing a new temporal operator ∇ that allows us to formalize the "history" of until, i.e., the "internal" universal quantification over the time instants between the current one and w'. This approach provides the basis for formalizing deduction systems for temporal logics endowed with the until operator. For concreteness, we give here a labeled natural deduction system for a linear-time logic endowed with the new history operator and show that, via a proper translation, such a system is also sound and complete with respect to the linear temporal logic *LTL* with until.

Keywords: Temporal logic, Until, LTL, Labeled Deduction, Natural Deduction.

1 Introduction

Until is a notoriously difficult temporal operator. This is because of its "ambivalent" nature of being an operator that is both existential and universal at the same time: $A\mathsf{U}B$ holds at the current time instant (sometimes "world" or "state" is used in place of "time instant") w iff either B holds at w or there *exists* a time instant w' in the future at which B holds and such that A holds in *all* the time instants between the current one and w'. The words in emphasis highlight the dual existential and universal nature of U, which poses a significant challenge when attempting to give deduction rules for until, so that deduction systems for temporal logics either deliberately exclude until from the set of operators considered or devise clever ways to formalize reasoning about until. And even if one manages to give rules, these often come at the price of additional difficulties for, or even the impossibility of, proving useful metatheoretic properties, such as normalization or the subformula property. (This is even more so in the case of Hilbert-style axiomatizations, which provide axioms for until, but are not easily usable for proof construction.) See, for instance, [1,2,7,12,13,22], where techniques for formalizing suitable inference rules include

1571-0661
doi:10.1016/j.entcs.2010.04.014

introducing additional information (such as the use of a Skolem function $f(A\mathsf{U}B)$ to name the time instant where B begins to hold), or exploiting the standard recursive unfolding of until

$$A\mathsf{U}B \equiv B \vee (A \wedge \mathsf{X}(A\mathsf{U}B)) \tag{1}$$

which says that $A\mathsf{U}B$ iff either B holds or A holds and in the successor time instant (as expressed by the *next* operator X) we have again $A\mathsf{U}B$.

In this paper, in contrast, we make explicit the duality of until by introducing a new temporal operator ∇ that allows us to formalize the "history" of until, i.e., the fact that when we have $A\mathsf{U}B$ the formula A holds in all the time instants between the current one and the one where B holds. We express this "historic" universal quantification by means of ∇ with respect to the following intuitive translation:

$$A\mathsf{U}B \equiv B \vee \mathsf{F}(\mathsf{X}B \wedge \nabla A) \tag{2}$$

That is: $A\mathsf{U}B$ iff either B holds or there exists a time instant w' in the future (as expressed by the *sometime in the future* operator F) such that

- B holds in the successor time instant, and
- A holds in all the time instants between the current one and w' (included).

The latter conjunct is precisely what the *history* operator ∇ expresses.[1] This is better seen when introducing labeling: since ∇ actually quantifies over the time instants in an interval (delimited by the current instant and the one where the B of the until holds), we adopt a labeling discipline that is slightly different from the more customary one of labeled deduction.

The framework of *labeled deduction* has been successfully employed for several non-classical, and in particular modal and temporal, logics, e.g., [8,23,24], since labeling provides a clean and effective way of dealing with modalities and gives rise to deduction systems with good proof-theoretical properties. The basic idea is that labels allow one to explicitly encode additional information, of a semantic or proof-theoretical nature, that is otherwise implicit in the logic one wants to capture. So, for instance, instead of a formula A, one can consider the *labeled formula* $b : A$, which intuitively means that A holds at the time instant denoted by b within the underlying Kripke semantics. One can also use labels to specify how time instants are related, e.g., the *relational formula* bRc states that the time instant c is accessible from b.

Considering labels that consist of a single time instant is not enough for ∇, as the operator is explicitly designed to speak about a sequence of time instants (namely, the ones constituting the history of the corresponding until, if indeed ∇ results from the translation of an U). We thus consider labels that are built out of a sequence of time instants, so that we can write $\alpha b_1 b_3 : \nabla A$ to express, intuitively, that A holds in the interval between time instants b_1 and b_3, which together with the sub-sequence α constitute a sequence of time instants $\alpha b_1 b_3$. This allows us to

[1] This is in contrast to the unfolding (1). The decoupling of U that we achieve with ∇ is precisely what allows us to give well-behaved (in a sense made clearer below) natural deduction rules.

give the natural deduction elimination rule

$$\frac{\alpha b_1 b_3 : \nabla A \quad b_1 \leqslant b_2 \quad b_2 \leqslant b_3}{\alpha b_1 b_2 : A}\ \nabla E$$

that says that if ∇A holds at time instant b_3 at the end of the sequence $\alpha b_1 b_3$ and if b_2 is in-between b_1 and b_3, as expressed by the relational formulas with the accessibility relation $\leqslant$, then we can conclude that A holds at b_2.

Dually, we can introduce ∇A at time instant b_3 at the end of the sequence $\alpha b_1 b_3$ whenever from the assumptions $b_1 \leqslant b_2$ and $b_2 \leqslant b_3$ for a *fresh* b_2 we can infer $\alpha b_1 b_2 : A$, i.e.,[2]

$$\begin{array}{c}[b_1 \leqslant b_2]\,[b_2 \leqslant b_3]\\ \vdots \\ \dfrac{\alpha b_1 b_2 : A}{\alpha b_1 b_3 : \nabla A}\ \nabla I\end{array}$$

The adoption of time instant sequences for labels has thus allowed us to give rules for ∇ that are well-behaved in the spirit of natural deduction [19]: there is precisely one introduction and one elimination rule for ∇, as well as for the other connectives and temporal operators ($\supset$, G, and X). This paves the way to a proof-theoretical analysis of the resulting natural deduction systems, e.g., to show proof normalization and other useful meta-theoretical analysis, which we are tackling in current work.

Moreover, the rules ∇I and ∇E provide a clean-cut way of reasoning about until, according to the translation (2), provided that we also give rules for F and X. These operators have a local nature, in the sense that they speak not about sequences of time instants but about single time instants. Still, we can easily give natural deduction rules for them by generalizing the more standard "single-time instant" rules (e.g., [1,2,12,18,23,24,25]) using our labeling with sequences of time instants. As we will discuss in more detail below, if we collapse the sequences of time instants to consider only the final time instant in the sequence (or, equivalently, if we simply ignore all the instants in a sequence but the last), then these rules reduce to the standard ones. For instance, for the *always in the future* operator G (the dual of F) and X, with the corresponding *successor relation* $\lhd$, we can give the elimination rules

$$\frac{\alpha b_1 : \mathsf{G}A \quad b_1 \leqslant b_2}{\alpha b_1 b_2 : A}\ \mathsf{G}E \qquad \text{and} \qquad \frac{\alpha b_1 : \mathsf{X}A \quad b_1 \lhd b_2}{\alpha b_1 b_2 : A}\ \mathsf{X}E$$

The rule $\mathsf{G}E$ says that if $\mathsf{G}A$ holds at time instant b_1, which is the last in the sequence αb_1 and b_2 is $\leqslant$-accessible from b_1 (i.e., $b_1 \leqslant b_2$), then we can conclude that A holds for the sequence $\alpha b_1 b_2$. The rule $\mathsf{X}E$ is justified similarly (via $\lhd$). The corresponding introduction rules are given in Section 4, together with rules for $\perp$ and the connective $\supset$, as well as a rule for induction on the underlying linear ordering. As we will see, we also need rules expressing the properties of the relations $\leqslant$ and $\lhd$. Moreover, the fact that we consider sequences of time instants as labels

[2] The side condition that b_2 is fresh means that b_2 is different from b_1 and b_3, and does not occur in any assumption on which $\alpha b_1 b_2 : A$ depends other than the discarded assumptions $b_1 \leqslant b_2$ and $b_2 \leqslant b_3$.

requires us to consider some structural rules to express properties of such sequences (with respect to formulas).

This approach thus provides the basis for formalizing deduction systems for temporal logics endowed with the until operator. For concreteness, we give here a labeled natural deduction system for a linear-time logic endowed with the new history operator ∇ and show that, via a proper translation, such a system is also sound and complete with respect to the linear temporal logic *LTL* with until. (We do not consider past explicitly here, but adding operators and rules for it should be unproblematic, e.g., as in [25].)

We proceed as follows. In Section 2, we briefly recall the syntax and semantics of *LTL*, and an axiomatization for it. In Section 3, we define LTL_∇, the logic that is obtained from *LTL* by replacing U with the history ∇, and give a validity-preserving translation, based on (2), from *LTL* into LTL_∇. In Section 4, we give a labeled natural deduction system $\mathcal{N}(LTL_\nabla)$ that it is sound with respect to the semantics of LTL_∇. By focusing only on those derivations whose conclusion and open assumptions correspond to the translation of *LTL*-formulas, we show that $\mathcal{N}(LTL_\nabla)$ can be used to capture reasoning in *LTL* and that it is in fact sound and complete with respect to the semantics of *LTL*. In Section 5, we draw conclusions and illustrate directions of current and future work. Due to lack of space, some of the proofs are only sketched here. Full proofs are given in [17].

2 The Linear Temporal Logic *LTL*

We recall the syntax and semantics of *LTL* and an axiomatization for it.

Definition 2.1 *Given a set $\mathcal{P}$ of propositional symbols, the set of* (well-formed) *LTL*-formulas *is defined by the grammar*

$$A ::= p \mid \perp \mid A \supset A \mid \mathsf{G}A \mid \mathsf{X}A \mid A\mathsf{U}A$$

where $p \in \mathcal{P}$. The set of LTL-atomic formulas *is $\mathcal{P} \cup \{\perp\}$. The* complexity *of an LTL-formula is the number of occurrences of the connective $\supset$ and of the temporal operators G, X, and U.*

The intuitive meaning of G, X, and U is the standard one: $\mathsf{G}A$ states that A holds always in the future, $\mathsf{X}A$ states that A holds in the next time instant, and $A\mathsf{U}B$ states that B holds at the current time instant or there is a time instant w in the future such that B holds in w and A holds in all the time instants between the current one and w. As usual, we can introduce abbreviations and use, e.g., $\neg$, $\vee$ and $\wedge$ for negation, disjunction, and conjunction, respectively: $\neg A \equiv A \supset \perp$, $A \vee B \equiv \neg A \supset B$, and $A \wedge B \equiv \neg(\neg A \vee \neg B)$. We can also define other temporal operators, e.g., $\mathsf{F}A \equiv \neg\mathsf{G}\neg A$ to express that A holds sometime in the future. We write Λ to denote a *set of LTL-formulas.*

Definition 2.2 *Let $\mathcal{N} = \langle \mathbb{N}, s : \mathbb{N} \to \mathbb{N}, \leq \rangle$ be the standard structure of natural numbers, where s and $\leq$ are respectively the successor function and the total (re-*

flexive) order relation. An LTL-model *is a pair* $\mathcal{M} = \langle \mathcal{N}, \mathcal{V} \rangle$ *where* $\mathcal{V} : \mathbb{N} \to 2^{\mathcal{P}}$*.* Truth *for an LTL-formula at a point* $n \in \mathbb{N}$ *in an LTL-model* $\mathcal{M} = \langle \mathcal{N}, \mathcal{V} \rangle$ *is the smallest relation* $\models_{LTL}$ *satisfying:*

$$
\begin{array}{rcl}
\mathcal{M}, n \models_{LTL} p & \textit{iff} & p \in \mathcal{V}(n) \\
\mathcal{M}, n \models_{LTL} A \supset B & \textit{iff} & \mathcal{M}, n \models_{LTL} A \textit{ implies } \mathcal{M}, n \models_{LTL} B \\
\mathcal{M}, n \models_{LTL} \mathsf{G}A & \textit{iff} & \mathcal{M}, m \models_{LTL} A \textit{ for all } m \geq n \\
\mathcal{M}, n \models_{LTL} \mathsf{X}A & \textit{iff} & \mathcal{M}, n+1 \models_{LTL} A \\
\mathcal{M}, n \models_{LTL} A\mathsf{U}B & \textit{iff} & \textit{there exists } n' \geq n \textit{ such that } \mathcal{M}, n' \models_{LTL} B \\
 & & \textit{and } \mathcal{M}, m \models_{LTL} A \textit{ for all } n \leq m < n'
\end{array}
$$

Note that $\mathcal{M}, n \not\models_{LTL} \bot$ *for every* $\mathcal{M}$ *and* n*. By extension, we write:*

$$
\begin{array}{rcl}
\mathcal{M} \models_{LTL} A & \textit{iff} & \mathcal{M}, n \models_{LTL} A \textit{ for every natural number } n \\
\mathcal{M} \models_{LTL} \Lambda & \textit{iff} & \mathcal{M} \models_{LTL} A \textit{ for all } A \in \Lambda \\
\Lambda \models_{LTL} A & \textit{iff} & \mathcal{M} \models_{LTL} \Lambda \textit{ implies } \mathcal{M} \models_{LTL} A\textit{, for every LTL-model } \mathcal{M}
\end{array}
$$

We now present a sound and complete Hilbert-style axiomatization, which we call $\mathcal{H}(LTL)$, for LTL (see, e.g., [10]). $\mathcal{H}(LTL)$ consists of the axioms

$$
\begin{array}{llll}
(A1) & \text{Any tautology instance} & (A2) & \mathsf{G}(A \supset B) \supset (\mathsf{G}A \supset \mathsf{G}B) \\
(A3) & (\mathsf{X}\neg A \leftrightarrow \neg \mathsf{X}A) & (A4) & \mathsf{X}(A \supset B) \supset (\mathsf{X}A \supset \mathsf{X}B) \\
(A5) & \mathsf{G}A \supset A \wedge \mathsf{XG}A & (A6) & \mathsf{G}(A \supset \mathsf{X}A) \supset (A \supset \mathsf{G}A) \\
(A7) & A\mathsf{U}B \leftrightarrow (B \vee (A \wedge \mathsf{X}(A\mathsf{U}B))) & (A8) & A\mathsf{U}B \supset \mathsf{F}B
\end{array}
$$

where we denote with $\leftrightarrow$ the double implication, and of the rules of inference

(MP) If A and $A \supset B$ then B $\quad (Nec_X)$ If A then $\mathsf{X}A$ $\quad (Nec_G)$ If A then $\mathsf{G}A$

The set of theorems of $\mathcal{H}(LTL)$ is the smallest set containing these axioms and closed with respect to these rules of inference.

3 LTL_∇: LTL with history

In this section, we give the linear temporal logic LTL_∇, which is obtained from LTL by replacing the operator U with a new unary temporal operator ∇, called *history*. The definition of the semantics of LTL_∇ requires a notion of truth given with respect to sequences of time instants rather than just to time instants. We will then provide a translation from the language of LTL into the language of LTL_∇ and show some properties of such a translation.

3.1 Syntax and semantics

Definition 3.1 *Given a set* $\mathcal{P}$ *of propositional symbols, the set of* (well-formed) LTL_∇-formulas *is defined by the grammar*

$$A ::= p \mid \bot \mid A \supset A \mid \mathsf{G}A \mid \mathsf{X}A \mid \nabla A$$

where $p \in \mathcal{P}$. The set of LTL_∇-atomic formulas *is $\mathcal{P} \cup \{\bot\}$. The* complexity *of an LTL_∇-formula is the number of occurrences of the connective $\supset$ and of the temporal operators* X, G, *and* ∇.

The intuitive meaning of the operators G and X is the same as for *LTL*, while ∇A intuitively states that A holds at any instant of a particular time interval (but here we see that we need sequences of time instants to formalize the semantics of the history operator, as we anticipated in the introduction). Again, we can define other connectives and operators as abbreviations, e.g., $\neg$, $\vee$, $\wedge$, F and so on. We write Γ to denote a *set of LTL_∇-formulas*.

To define a labeled deduction system for the logic LTL_∇, we extend the language with a set of labels and finite sequences of labels, and introduce the notions of labeled formula and relational formula.

Definition 3.2 *Let $\mathcal{L}$ be a set of labels. A finite non-empty sequence of labels (namely, an element of $\mathcal{L}^+$) is called a* sequence. *If A is an LTL_∇-formula and α is a sequence, then $\alpha : A$ is a* labeled (well-formed) formula *(*lwff *for short). The set of* relational (well-formed) formulas *(*rwffs *for short) is the set of expressions of the form $b \leqslant c$ or $b \lhd c$, where b and c are labels.*

In the rest of the paper, we will assume given a fixed denumerable set $\mathcal{L}$ of labels and we will use $b, c, d, \ldots$ to denote labels, α, β, γ to denote finite sequences of labels[3] (e.g., $bcd \ldots$ or just b in the case of a sequence consisting of only one time instant), φ to denote a *generic formula* (either labeled or relational) and Φ to denote a *set of generic formulas*.

Definition 3.3 *An* observation sequence *is a non-empty sequence $\sigma = [n_0, \ldots, n_k]$ of natural numbers.* Truth *for an LTL_∇-formula at an observation sequence σ in an LTL-model $\mathcal{M} = \langle \mathcal{N}, \mathcal{V} \rangle$ is the smallest relation $\models_\triangledown$ satisfying:*

$$
\begin{array}{rcl}
\mathcal{M}, [n_0, \ldots, n_k] \models_\triangledown p & \textit{iff} & p \in \mathcal{V}(n_k) \\
\mathcal{M}, [n_0, \ldots, n_k] \models_\triangledown A \supset B & \textit{iff} & \mathcal{M}, [n_0, \ldots, n_k] \models_\triangledown A \textit{ implies} \\
 & & \mathcal{M}, [n_0, \ldots, n_k] \models_\triangledown B \\
\mathcal{M}, [n_0, \ldots, n_k] \models_\triangledown \mathsf{G}A & \textit{iff} & \mathcal{M}, [n_0, \ldots, n_k, m] \models_\triangledown A \textit{ for all } m \geq n_k \\
\mathcal{M}, [n_0, \ldots, n_k] \models_\triangledown \mathsf{X}A & \textit{iff} & \mathcal{M}, [n_0, \ldots, n_k, n_k + 1] \models_\triangledown A \\
\mathcal{M}, [n_0, \ldots, n_{k-1}, n_k] \models_\triangledown \nabla A & \textit{iff} & \mathcal{M}, [n_0, \ldots, n_{k-1}, m] \models_\triangledown A \\
 & & \textit{for all } n_{k-1} \leq m \leq n_k \ (\textit{if } 0 < k) \\
\mathcal{M}, [n_0] \models_\triangledown \nabla A & \textit{iff} & \mathcal{M}, [n_0] \models_\triangledown A
\end{array}
$$

By extension, we write:

$$
\begin{array}{rcl}
\mathcal{M} \models_\triangledown A & \textit{iff} & \mathcal{M}, \sigma \models_\triangledown A \textit{ for every observation sequence } \sigma \\
\mathcal{M} \models_\triangledown \Gamma & \textit{iff} & \mathcal{M} \models_\triangledown A \textit{ for all } A \in \Gamma \\
\Gamma \models_\triangledown A & \textit{iff} & \mathcal{M} \models_\triangledown \Gamma \textit{ implies } \mathcal{M} \models_\triangledown A\textit{, for every LTL-model } \mathcal{M}
\end{array}
$$

[3] With a slight abuse of notation, we will also use α, β, γ to denote possibly empty subsequences and thus write $\alpha b_1 \ldots b_k$ (for $k \geq 1$) to denote a sequence where α may be empty.

Given an LTL-model $\mathcal{M}$, *a* structure *is a pair* $\mathcal{S} = \langle \mathcal{M}, \mathcal{I} \rangle$ *where* $\mathcal{I} : \mathcal{L} \to \mathbb{N}$. *Let* Σ *be the set of observation sequences and* $\mathcal{I}^+ : \mathcal{L}^+ \to \Sigma$ *the extension of* $\mathcal{I}$ *to sequences, i.e.,* $\mathcal{I}^+(b_0 \dots b_n) = [\mathcal{I}(b_0), \dots, \mathcal{I}(b_n)]$. Truth *for a generic formula* φ *in a structure* $\mathcal{S} = \langle \mathcal{M}, \mathcal{I} \rangle$ *is the smallest relation* $\models_\triangledown$ *satisfying:*

$$\begin{array}{lll} \mathcal{M}, \mathcal{I} \models_\triangledown a \leqslant b & \textit{iff} & \mathcal{I}(a) \leq \mathcal{I}(b) \\ \mathcal{M}, \mathcal{I} \models_\triangledown a \lhd b & \textit{iff} & \mathcal{I}(b) = \mathcal{I}(a) + 1 \\ \mathcal{M}, \mathcal{I} \models_\triangledown \alpha : A & \textit{iff} & \mathcal{M}, \mathcal{I}^+(\alpha) \models_\triangledown A \end{array}$$

Note that $\mathcal{M}, \sigma \not\models_\triangledown \bot$ *and* $\mathcal{M}, \mathcal{I} \not\models_\triangledown \alpha : \bot$ *for every* $\mathcal{M}$, σ *and* $\mathcal{I}$.

Given a set Φ *of generic formulas and a generic formula* φ*:*

$$\begin{array}{rll} \mathcal{M}, \mathcal{I} \models_\triangledown \Phi & \textit{iff} & \mathcal{M}, \mathcal{I} \models_\triangledown \varphi \textit{ for all } \varphi \in \Phi \\ \Phi \models_\triangledown \varphi & \textit{iff} & \mathcal{M}, \mathcal{I} \models_\triangledown \Phi \textit{ implies } \mathcal{M}, \mathcal{I} \models_\triangledown \varphi \textit{ for all } \mathcal{M} \textit{ and } \mathcal{I} \end{array}$$

3.2 A translation from LTL into $LTL_\triangledown$

LTL and $LTL_\triangledown$ are, obviously, related logics. In fact, below we will define a validity-preserving translation $(\cdot)^*$ from LTL into $LTL_\triangledown$. Then, in Lemma 3.5, we will show that if an $LTL_\triangledown$-formula corresponds to the translation of some LTL-formula, then it can be interpreted "locally", i.e., its truth value with respect to an observation sequence depends only on the last element of the sequence. Finally, in Lemma 3.7 and Theorem 3.8, we will use this result to prove that the translation preserves the validity of formulas. This property allows us to use the deduction system for $LTL_\triangledown$, which will be presented in Section 4, for reasoning on LTL too, as we will show in Section 4.2, when discussing soundness and completeness of the system.

Definition 3.4 *We define the translation* $(\cdot)^*$ *from the language of LTL into the language of* $LTL_\triangledown$ *inductively as follows:*

$$\begin{array}{lllllll} (p)^* & = & p\,, \textit{ for } p \textit{ atomic} & & (\mathsf{G}A)^* & = & \mathsf{G}\,(A)^* \\ (\bot)^* & = & \bot & & (\mathsf{X}A)^* & = & \mathsf{X}\,(A)^* \\ (A \supset B)^* & = & (A)^* \supset (B)^* & & (A\mathsf{U}B)^* & = & (B)^* \vee (\,\mathsf{F}\,(\,\mathsf{X}\,(B)^* \wedge \nabla(A)^*\,)) \end{array}$$

We extend $(\cdot)^*$ *to sets of formulas in the obvious way:* $\Lambda^* = \{B^* \mid B \in \Lambda\}$.

Lemma 3.5 *Let* $\mathcal{M}$ *be an LTL-model,* $[n_1, \dots, n_k]$ *an observation sequence, and* A *an LTL-formula. Then* $\mathcal{M}, [n_1, \dots, n_k] \models_\triangledown A^*$ *iff* $\mathcal{M}, [m_1, \dots, m_r, n_k] \models_\triangledown A^*$ *for every sequence* $m_1, \dots, m_r$.

Corollary 3.6 *Let* $\mathcal{M}$ *be an LTL-model,* $[n_1, \dots, n_k]$ *an observation sequence, and* A *an LTL-formula. Then* $\mathcal{M}, [n_1, \dots, n_k] \models_\triangledown A^*$ *iff* $\mathcal{M}, [n_k] \models_\triangledown A^*$.

Lemma 3.7 *Let* $\mathcal{M}$ *be an LTL-model,* n *a natural number, and* A *an LTL-formula. Then* $\mathcal{M}, n \models_{LTL} A$ *iff* $\mathcal{M}, [n] \models_\triangledown A^*$.

Theorem 3.8 *Let* Λ *be a set of LTL-formulas,* A *an LTL-formula and* $\Lambda^* = \{B^* \mid B \in \Lambda\}$. *Then* $\Lambda \models_{LTL} A$ *iff* $\Lambda^* \models_\triangledown A^*$.

Proof By Definition 2.2, $\Lambda \models_{LTL} A$ iff $\forall \mathcal{M}.\, \mathcal{M} \models_{LTL} \Lambda$ implies $\mathcal{M} \models_{LTL} A$ iff $\forall \mathcal{M}.\, (\forall B \in \Lambda.\, \forall n.\, \mathcal{M}, n \models_{LTL} B$ implies $\forall n.\, \mathcal{M}, n \models_{LTL} A\,)$ iff (by Lemma 3.7)

$\forall\mathcal{M}.\,(\forall B \in \Lambda.\,\forall n.\,\mathcal{M},[n] \models_{\triangledown} B^*$ implies $\forall n.\,\mathcal{M},[n] \models_{\triangledown} A^*)$ iff (by Lemma 3.5) $\forall\mathcal{M}.\,(\forall B \in \Lambda.\,\forall\sigma.\,\mathcal{M},\sigma \models_{\triangledown} B^*$ implies $\forall\sigma.\,\mathcal{M},\sigma \models_{\triangledown} A^*)$ iff (by Definition 3.3) $\forall\mathcal{M}.\,(\forall B \in \Lambda.\,\mathcal{M} \models_{\triangledown} B^*$ implies $\mathcal{M} \models_{\triangledown} A^*)$ iff $\forall\mathcal{M}.\,(\mathcal{M} \models_{\triangledown} \Lambda^*$ implies $\mathcal{M} \models_{\triangledown} A^*)$ iff $\Lambda^* \models_{\triangledown} A^*$. □

4 $\mathcal{N}(LTL_\triangledown)$: a labeled natural deduction system for $LTL_\triangledown$

In this section, we will first define a labeled natural deduction system $\mathcal{N}(LTL_\triangledown)$ that is sound with respect to the semantics of $LTL_\triangledown$. Then, by considering a restriction of the set of $\mathcal{N}(LTL_\triangledown)$-derivations and by using the translation $(\cdot)^*$ and the related results, we will show that $\mathcal{N}(LTL_\triangledown)$ can be also used for reasoning on LTL: we will prove soundness with respect to the semantics of LTL and we will give a proof of weak completeness with respect to LTL, by exploiting the Hilbert-style axiomatization $\mathcal{H}(LTL)$.

4.1 The rules of $\mathcal{N}(LTL_\triangledown)$

The rules of $\mathcal{N}(LTL_\triangledown)$ are given in Figure 1. In $\mathcal{N}(LTL_\triangledown)$ we do not make use of a proper relational labeling algebra (as, e.g., in [24]) that contains rules that derive rwffs from other rwffs or even lwffs. Since we are mainly interested in the derivation of logical formulas, we rather follow an approach that aims at simplifying the system: we use rwffs only as assumptions for the derivation of lwffs (as in Simpson's system for intuitionistic modal logic [23]). Thus, in $\mathcal{N}(LTL_\triangledown)$ there are no rules whose conclusion is an rwff.

The rules $\supset I$ and $\supset E$ are just the labeled version of the standard [19] natural deduction rules for implication introduction and elimination, where the notion of *discharged/open assumption* is also standard; e.g., $[\alpha : A]$ means that the formula is discharged in $\supset I$. The rule $\perp E$ is a labeled version of *reductio ad absurdum*, where we do not constrain the time instant sequence (α_2) in which we derive a contradiction to be the same (α_1) as in the assumption.

The rules for the introduction and the elimination of G and X share the same structure since they both have a "universal" formulation. Consider, for instance, G and the corresponding relation $\leqslant$. The idea underlying the introduction rule $\mathsf{G}I$ is that the meaning of $\alpha b_1 : \mathsf{G}A$ is given by the metalevel implication $b_1 \leqslant b_2 \Longrightarrow \alpha b_1 b_2 : A$ for an arbitrary b_2 $\leqslant$-accessible from b_1 (where the arbitrariness of b_2 is ensured by the side-condition on the rule). As we remarked above, the operators G and X have a local nature, in that when we write $\alpha b_1 : \mathsf{G}A$ (and similarly for $\alpha b_1 : \mathsf{X}A$) we are stating that $\mathsf{G}A$ holds at time instant b_1, which is the last in the sequence αb_1. Hence, the elimination rule $\mathsf{G}E$ says that if b_2 is $\leqslant$-accessible from b_1 (i.e., $b_1 \leqslant b_2$), then we can conclude that A holds for the sequence $\alpha b_1 b_2$. Similar observations hold for X and the corresponding relation $\lhd$.

The rule $ser_\lhd$ models the fact that every time instant has an immediate successor, while the rule $lin_\lhd$ specifies that such a successor must be unique. $ser_\lhd$ tells us that if assuming $b_1 \lhd b_2$ we can derive $\alpha : A$, then we can discharge the assumption and conclude that indeed $\alpha : A$. $lin_\lhd$ is slightly more complex: assume that b_1

$$\begin{array}{c}[\alpha_1 : A \supset \bot]\\ \vdots \\ \dfrac{\alpha_2 : \bot}{\alpha_1 : A}\ \bot E\end{array} \qquad \begin{array}{c}[\alpha : A]\\ \vdots \\ \dfrac{\alpha : B}{\alpha : A \supset B}\ \supset I\end{array} \qquad \dfrac{\alpha : A \supset B \quad \alpha : A}{\alpha : B}\ \supset E$$

$$\begin{array}{c}[b_1 \leqslant b_2]\\ \vdots \\ \dfrac{\alpha b_1 b_2 : A}{\alpha b_1 : \mathsf{G}A}\ \mathsf{G}I\end{array} \qquad \dfrac{\alpha b_1 : \mathsf{G}A \quad b_1 \leqslant b_2}{\alpha b_1 b_2 : A}\ \mathsf{G}E \qquad \begin{array}{c}[b_1 \lhd b_2]\\ \vdots \\ \dfrac{\alpha : A}{\alpha : A}\ \mathit{ser}_{\lhd}\end{array} \qquad \dfrac{b_1 \lhd b_2 \quad b_1 \lhd b_3 \quad \varphi \quad \begin{array}{c}[\varphi[b_3/b_2]]\\ \vdots \\ \alpha : A\end{array}}{\alpha : A}\ \mathit{lin}_{\lhd}$$

$$\begin{array}{c}[b_1 \lhd b_2]\\ \vdots \\ \dfrac{\alpha b_1 b_2 : A}{\alpha b_1 : \mathsf{X}A}\ \mathsf{X}I\end{array} \qquad \dfrac{\alpha b_1 : \mathsf{X}A \quad b_1 \lhd b_2}{\alpha b_1 b_2 : A}\ \mathsf{X}E \qquad \begin{array}{c}[b_1 \leqslant b_1]\\ \vdots \\ \dfrac{\alpha : A}{\alpha : A}\ \mathit{refl}_{\leqslant}\end{array} \qquad \dfrac{b_1 \leqslant b_2 \quad b_2 \leqslant b_3 \quad \begin{array}{c}[b_1 \leqslant b_3]\\ \vdots \\ \alpha : A\end{array}}{\alpha : A}\ \mathit{trans}_{\leqslant}$$

$$\begin{array}{c}[b_1 \leqslant b_2]\,[b_2 \leqslant b_3]\\ \vdots \\ \dfrac{\alpha b_1 b_2 : A}{\alpha b_1 b_3 : \nabla A}\ \nabla I\end{array} \qquad \dfrac{\alpha b_1 b_3 : \nabla A \quad b_1 \leqslant b_2 \quad b_2 \leqslant b_3}{\alpha b_1 b_2 : A}\ \nabla E \qquad \dfrac{\beta b : A^l}{\alpha b : A^l}\ \mathit{last}$$

$$\dfrac{b_1 \leqslant b_2 \quad b_2 \leqslant b_1 \quad \alpha b_1 : A}{\alpha b_2 : A}\ \mathit{eq}_{\leqslant} \qquad \dfrac{b_1 \leqslant b_2 \quad \varphi \quad \begin{array}{c}[\varphi[b_2/b_1]]\\ \vdots \\ \alpha : A\end{array} \quad \begin{array}{c}[b_1 \lhd b']\,[b' \leqslant b_2]\\ \vdots \\ \alpha : A\end{array}}{\alpha : A}\ \mathit{split}_{\leqslant}$$

$$\dfrac{b_1 \lhd b_2 \quad \begin{array}{c}[b_1 \leqslant b_2]\\ \vdots \\ \alpha : A\end{array}}{\alpha : A}\ \mathit{base}_{\leqslant} \qquad \dfrac{\alpha b_0 : A \quad b_0 \leqslant b \quad \begin{array}{c}[b_0 \leqslant b_i]\,[b_i \lhd b_j]\,[\alpha b_i : A]\\ \vdots \\ \alpha b_j : A\end{array}}{\alpha b : A}\ \mathit{ind}$$

The rules have the following side conditions:

- In $\mathsf{X}I$ ($\mathsf{G}I$), b_2 is *fresh*, i.e., it is different from b_1 and does not occur in any assumption on which $\alpha b_1 b_2 : A$ depends other than the discarded assumption $b_1 \lhd b_2$ ($b_1 \leqslant b_2$).
- In ∇I, b_2 is *fresh*, i.e., it is different from b_1 and b_3, and does not occur in any assumption on which $\alpha b_1 b_2 : A$ depends other than the discarded assumptions $b_1 \leqslant b_2$ and $b_2 \leqslant b_3$.
- In *last*, the formula must be of the form A^l, as defined in (3).
- In $\mathit{ser}_{\lhd}$, b_2 is fresh, i.e., it is different from b and does not occur in any assumption on which $\alpha : A$ depends other than the discarded assumption $b_1 \lhd b_2$.
- In $\mathit{split}_{\leqslant}$, b' is fresh, i.e., it is different from b_1 and b_2 and does not occur in any assumption on which $\alpha : A$ depends other than the discarded assumptions $b_1 \lhd b'$ and $b' \leqslant b_2$.
- In *ind*, b_i and b_j are fresh, i.e., they are different from each other and from b and b_0, and do not occur in any assumption on which $\alpha b_0 b_j : A$ depends other than the discarded assumptions of the rule.

Figure 1. The rules of $\mathcal{N}(LTL_{\nabla})$

had two different immediate successors b_2 and b_3 (which we know cannot be) and assume that the generic formula φ holds; if by substituting b_3 for b_2 in φ we obtain $\alpha : A$, then we can discharge the assumption and conclude that indeed $\alpha : A$.

Similarly, the rules $\mathit{refl}_{\leqslant}$ and $\mathit{trans}_{\leqslant}$ state the reflexivity and transitivity of $\leqslant$, while $\mathit{eq}_{\leqslant}$ captures substitution of equals. [4] The rule $\mathit{split}_{\leqslant}$ states that if $b_1 \leqslant b_2$, then either $b_1 = b_2$ or $b_1 < b_2$. The rule thus works in the style of a disjunction elimination: if by assuming either of the two cases, we can derive a formula $\alpha : A$,

[4] Recall that in this paper we use rwffs only as assumptions for the derivation of lwffs, so we do not need a more general rule that concludes $\varphi[b_2/b_1]$ from φ, $b_1 \leqslant b_2$ and $b_2 \leqslant b_1$.

then we can discharge the assumptions and conclude $\alpha : A$. Since we do not use $=$ and $<$ explicitly in our syntax, we express such relations in an indirect way: the equality of b_1 and b_2 is expressed by replacing one with the other in a generic formula φ, $<$ by the composition of $\lhd$ and $\leqslant$.

The rule $base_{\leqslant}$ expresses the fact that $\leqslant$ contains $\lhd$, while the rule *ind* models the induction principle underlying the relation between $\lhd$ and $\leqslant$. If (base case) A holds in αb_0 and if (inductive step) by assuming that A holds in αb_i for an arbitrary b_i $\leqslant$-accessible from b_0, we can derive that A holds also in αb_j, where b_j is the immediate successor of b_i, then we can conclude that A holds in every αb such that b is $\leqslant$-accessible from b_0.[5]

Finally, we have three rules that speak about the history and the label sequences: the rules ∇I and ∇E, which we already described in the introduction, and *last*. This rule expresses what we also anticipated in the introduction: the standard operators (and connectives) of *LTL* speak not about sequences of time instants but about single time instants, and thus if a formula A whose outermost operator is not ∇ holds at βb, then we can safely replace β by any other sequence α and conclude that A holds at αb. To formalize this, we define the set of (well-formed) LTL^l-*formulas* (denoted by A^l) by means of the grammar

$$\begin{array}{rcl} A^l & ::= & p \,|\perp|\, (A^l) \supset (A^l) \mid \mathsf{G}(A^{l\nabla}) \mid \mathsf{X}(A^{l\nabla}) \\ A^{l\nabla} & ::= & A^l \mid (A^{l\nabla}) \supset (A^{l\nabla}) \mid \nabla(A^{l\nabla}) \end{array} \tag{3}$$

where p is a propositional symbol. Hence, in a formula A^l, the history operator ∇ can only appear in the scope of a temporal operator G (and thus of F as in the translation (2)) or X. The rule *last* applies to these formulas only; in fact, the "l" in A^l stands for "last", but it also conveniently evokes both "local" and "*LTL*". For formulas ∇A whose outermost operator is the history operator ∇, such a rule does not make sense (and in fact is not sound) as it would mean changing the interval over which A holds.

Such considerations are formalized in the following lemma, where we prove, for LTL^l-formulas, a result that is the analogous of the one given in Lemma 3.5 with respect to the translation of *LTL*-formulas.[6] At the same time, we also prove that if A is a formula belonging to the syntactic category $A^{l\nabla}$ of the grammar (3) (we will call such formulas $LTL^{l\nabla}$-*formulas*), then the truth value of A depends on at most the last two elements of an observation sequence.

Lemma 4.1 *Let $\mathcal{M}$ be an LTL-model, $[n_1, \dots, n_k]$ an observation sequence, A^l an LTL^l-formula and $A^{l\nabla}$ an $LTL^{l\nabla}$-formula. Then: (i) $\mathcal{M}, [n_1, \dots, n_k] \models_\nabla A^l$ iff $\mathcal{M}, [m_1, \dots, m_r, n_k] \models_\nabla A^l$ for every sequence $m_1, \dots, m_r$ and (ii) $\mathcal{M}, [n_1, \dots, n_{k-1}, n_k] \models_\nabla A^{l\nabla}$ iff $\mathcal{M}, [m_1, \dots, m_r, n_{k-1}, n_k] \models_\nabla A^{l\nabla}$ for every sequence $m_1, \dots, m_r$.*

[5] The rule is given only in terms of relations between labels, since we restrict the treatment of operators in the system to the specific rules for their introduction and elimination.

[6] In fact, Lemma 3.5 is a direct consequence of Lemma 4.1 and of Lemma 4.3 below.

Given the rules in Fig. 1, the notions of *derivation*, *assumption* (*open* or *discharged*, as we remarked) and *conclusion* are the standard ones for natural deduction systems [19]. We write $\Phi \vdash_{\triangledown} \alpha : A$ to say that there exists a derivation of $\alpha : A$ in the system $\mathcal{N}(LTL_{\triangledown})$ whose open assumptions are all contained in the set of formulas Φ. A derivation of $\alpha : A$ in $\mathcal{N}(LTL_{\triangledown})$ where all the assumptions are discharged is a *proof* of $\alpha : A$ in $\mathcal{N}(LTL_{\triangledown})$ and we then say that $\alpha : A$ is a *theorem* of $\mathcal{N}(LTL_{\triangledown})$ and write $\vdash_{\triangledown} \alpha : A$.

To denote that Π is a derivation of $\alpha : A$ whose set of assumptions may contain the formulas $\varphi_1, \dots, \varphi_n$, we write

$$\begin{array}{c} \varphi_1 \dots \varphi_n \\ \Pi \\ \alpha : A \end{array}$$

If we are interested in LTL-reasoning, then we can restrict our attention to a subset of the $\mathcal{N}(LTL_{\triangledown})$-derivations, namely, to the derivations where the conclusion and all the open assumptions correspond to the translations of LTL-formulas.

Definition 4.2 *Let Π be a derivation in $\mathcal{N}(LTL_{\triangledown})$ and Φ the set containing the conclusion and the open assumptions of Π. We say that Π is an LTL-*derivation *iff there exists a label b such that for every φ in Φ there exists an LTL-formula A such that $\varphi = b : A^*$. We write $\Lambda \vdash_{LTL} A$ to denote that in $\mathcal{N}(LTL_{\triangledown})$ there exists an LTL-derivation of $b : A^*$ from open assumptions in a set Φ, where $\Lambda = \{B \mid b : B^* \in \Phi\}$.*

In Definition 4.2, we require all the open assumptions and the conclusion of an LTL-derivation to be lwffs labeled by the same single label b. Note that, as a consequence of Corollary 3.6, we would obtain the same notion of LTL-derivation by requiring instead that such formulas were labeled by the same sequence α.

In Section 4.2, we will show that $\mathcal{N}(LTL_{\triangledown})$ is sound with respect to the semantics of $LTL_{\triangledown}$ and, by considering the notion of LTL-derivability $\vdash_{LTL}$, that it is sound and weakly complete with respect to LTL. An investigation of completeness with respect to $LTL_{\triangledown}$ is left for future work, together with the formalization of an axiomatization of $LTL_{\triangledown}$.

Related to this, it is important to understand what exactly is the relationship of the class of LTL^l-formulas and the class of LTL-formulas, in particular with respect to the translation $(\cdot)^*$. It is not difficult to see that the co-domain of the translation is included in LTL^l by construction of $(\cdot)^*$, i.e., by induction on the formula complexity it follows that:

Lemma 4.3 *If A is an LTL-formula, then A^* is an LTL^l-formula.*

The other direction is trickier, as it basically amounts to defining an inverse translation. To solve this problem, we have been considering normal forms of LTL^l-formulas and we conjecture that the following fact indeed holds.

Conjecture 4.4 *If A is an LTL^l-formula, then there exists an LTL-formula B such that B^* is semantically equivalent to A.*

4.2 Soundness and completeness

Theorem 4.5 *For every set Φ of labeled and relational formulas and every labeled formula $\alpha : A$, if $\Phi \vdash_{\triangledown} \alpha : A$, then $\Phi \models_{\triangledown} \alpha : A$.*

Proof The proof proceeds by induction on the structure of the derivation of $\alpha : A$. The base case is when $\alpha : A \in \Phi$ and is trivial. There is one step case for every rule and we show here only the two representative cases

$$\dfrac{\begin{array}{c}[b_1 \leqslant b_2] \quad [b_2 \leqslant b_3] \\ \Pi \\ \beta b_1 b_2 : B\end{array}}{\beta b_1 b_3 : \nabla B}\ \nabla I \qquad \text{and} \qquad \dfrac{\begin{array}{c}\Pi \\ \beta' b : A\end{array}}{\beta b : A}\ \mathit{last}$$

Some more cases are in [17]. First, consider the case in which the last rule application is a ∇I, where $\alpha = \beta b_1 b_3$, $A = \nabla B$, and Π is a proof of $\beta b_1 b_2 : B$ from hypotheses in Φ', with b_2 fresh and with $\Phi' = \Phi \cup \{b_1 \leqslant b_2\} \cup \{b_2 \leqslant b_3\}$. By the induction hypothesis, for every interpretation $\mathcal{I}$, if $\mathcal{M}, \mathcal{I} \models_{\triangledown} \Phi'$, then $\mathcal{M}, \mathcal{I} \models_{\triangledown} \beta b_1 b_2 : B$. We let $\mathcal{I}$ be any interpretation such that $\mathcal{M}, \mathcal{I} \models_{\triangledown} \Phi$, and show that $\mathcal{M}, \mathcal{I} \models_{\triangledown} \beta b_1 b_3 : \nabla B$. Let $\mathcal{I}(b_1) = n$, $\mathcal{I}(b_3) = m$ and $\mathcal{I}^+(\beta) = [n_1, \ldots, n_k]$. Since b_2 is fresh, we can extend $\mathcal{I}$ to an interpretation (still called $\mathcal{I}$ for simplicity) such that $\mathcal{I}(b_2) = n + i$ for an arbitrary $0 \le i \le m$. The induction hypothesis yields $\mathcal{M}, \mathcal{I} \models_{\triangledown} \beta b_1 b_2 : B$, i.e., $\mathcal{M}, [n_1, \ldots, n_k, n, n+i] \models_{\triangledown} B$, and thus, since i is an arbitrary point between 0 and m, we obtain $\mathcal{M}, [n_1, \ldots, n_k, n, n+m] \models_{\triangledown} \nabla B$. It follows $\mathcal{M}, \mathcal{I} \models_{\triangledown} \beta b_1 b_3 : \nabla B$.

Now consider the case in which the last rule applied is *last* and $\alpha = \beta b$, where Π is a proof of $\beta' b : A$ from hypotheses in Φ. By applying the induction hypothesis on Π, we have $\Phi \models_{\triangledown} \beta' b : A$. We proceed by considering a generic *LTL*-model $\mathcal{M}$ and a generic interpretation $\mathcal{I}$ on it such that $\mathcal{M}, \mathcal{I} \models_{\triangledown} \Phi$ and showing that this entails $\mathcal{M}, \mathcal{I} \models_{\triangledown} \beta b : A$. By the induction hypothesis, $\mathcal{M}, \mathcal{I} \models_{\triangledown} \beta' b : A$, i.e., $\mathcal{M}, \mathcal{I}^+(\beta' b) \models_{\triangledown} A$ by Definition 3.3. Since A is an LTL^l-formula by the side condition of the rule and the two observation sequences $\mathcal{I}^+(\beta' b)$ and $\mathcal{I}^+(\beta b)$ share the same last element $\mathcal{I}(b)$, we can apply Lemma 4.1 and obtain $\mathcal{M}, \mathcal{I}^+(\beta b) \models_{\triangledown} A$, i.e., $\mathcal{M}, \mathcal{I} \models_{\triangledown} \beta b : A$ by Definition 3.3. □

By exploiting the translation of Section 3.2 and the notion of *LTL*-derivation of Definition 4.2, we also prove a result of soundness with respect to *LTL*.

Theorem 4.6 *For every set Λ of LTL-formulas and every LTL-formula A, if $\Lambda \vdash_{LTL} A$, then $\Lambda \models_{LTL} A$.*

Proof By definition of $\vdash_{LTL}$, for a given label b, there exists a derivation in $\mathcal{N}(LTL_{\triangledown})$ of $b : A^*$ from open assumptions in $\Phi = \{b : B^* \mid B \in \Lambda\}$. By Theorem 4.5, $\Phi \vdash_{\triangledown} b : A^*$ implies $\Phi \models_{\triangledown} b : A^*$. Since b is generic, we have that for every *LTL*-model $\mathcal{M}$ and every interpretation $\mathcal{I}$, $\mathcal{M}, \mathcal{I} \models_{\triangledown} \Phi$ implies $\mathcal{M}, \mathcal{I} \models_{\triangledown} b : A^*$ iff for every natural number n, $\mathcal{M}, [n] \models_{\triangledown} \Lambda^*$ implies $\mathcal{M}, [n] \models_{\triangledown} A^*$, where $\Lambda^* = \{B^* \mid B \in \Lambda\}$. By Lemma 3.5, we infer that for every observation sequence σ, $\mathcal{M}, \sigma \models_{\triangledown} \Lambda^*$ implies $\mathcal{M}, \sigma \models_{\triangledown} A^*$. By Definition 3.3, $\Lambda^* \models_{\triangledown} A^*$ and thus, by Theorem 3.8, we conclude $\Lambda \models_{LTL} A$. □

As we anticipated, an analysis of the completeness of $\mathcal{N}(LTL_\nabla)$ with respect to LTL_∇ is left for future work. Here we discuss completeness with respect to LTL. The proposed natural deduction system consists of only finitary rules; consequently, it cannot be strongly complete for LTL.[7] Nevertheless, by using the axiomatization $\mathcal{H}(LTL)$ and the translation $(\cdot)^*$, we can give a proof of weak completeness for it; namely:

Theorem 4.7 *For every LTL-formula A, if $\models_{LTL} A$, then $\vdash_{LTL} A$.*

Proof We can prove the theorem by showing that $\mathcal{N}(LTL_\nabla)$ is complete with respect to the axiomatization $\mathcal{H}(LTL)$ given in Section 2, which is sound and complete for the logic LTL. That is, we need to prove that: (i) the translation, via $(\cdot)^*$, of every axiom of $\mathcal{H}(LTL)$ is provable in $\mathcal{N}(LTL_\nabla)$ by means of an LTL-derivation, and (ii) the notion of $\vdash_{LTL}$ is closed under the (labeled equivalent of the) rules of inference of $\mathcal{H}(LTL)$. Showing (ii) is straightforward and we omit it here. With regard to (i), due to lack of space, we only give here a few examples. The other cases are presented in [17].

(*A2*)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[b:\mathsf{G}(A\supset B)]^1 \quad [b\leqslant c]^3}{bc:A\supset B}\,\mathsf{G}E \qquad \dfrac{[b:\mathsf{G}A]^2\quad [b\leqslant c]^3}{bc:A}\,\mathsf{G}E}{bc:B}\,\supset E}{b:\mathsf{G}B}\,\mathsf{G}I^3}{b:\mathsf{G}A\supset \mathsf{G}B}\,\supset I^2}{b:\mathsf{G}(A\supset B)\supset(\mathsf{G}A\supset\mathsf{G}B)}\,\supset I^1$$

(*A3*)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[b:\mathsf{X}\neg A]^1\quad[b\lhd c]^2}{bc:\neg A}\,\mathsf{X}E\qquad\dfrac{[b:\mathsf{X}A]^3\quad[b\lhd c]^2}{bc:A}\,\mathsf{X}E}{bc:\bot}\,\supset E}{b:\neg\mathsf{X}A}\,\bot E^3}{b:\neg\mathsf{X}A}\,ser^2_\lhd}{b:\mathsf{X}\neg A\supset\neg\mathsf{X}A}\,\supset I^1$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[b:\neg\mathsf{X}A]^1\qquad \dfrac{\dfrac{[b\lhd c]^2\quad[b\lhd d]^4\quad[bc:A]^3}{bd:A}\,lin_\lhd}{b:\mathsf{X}A}\,\mathsf{X}I^4}{b:\bot}\,\supset E}{bc:\neg A}\,\supset^3}{b:\mathsf{X}\neg A}\,\mathsf{X}I^2}{b:\neg\mathsf{X}A\supset\mathsf{X}\neg A}\,\supset I^1$$

(*A5*)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[b:\mathsf{G}A]^1\quad[b\leqslant b]^2}{bb:A}\,\mathsf{G}E}{b:A}\,last}{b:A}\,refl^2_\leqslant \qquad \dfrac{\dfrac{\dfrac{\dfrac{[b\lhd c]^3\qquad \dfrac{[b\leqslant c]^5\quad[c\leqslant d]^4\quad \dfrac{[b:\mathsf{G}A]^1\quad[b\leqslant d]^6}{bd:A}\,\mathsf{G}E}{bd:A}\,trans^6_\leqslant}{bd:A}\,base^5_\leqslant}{bcd:A}\,last}{bc:\mathsf{G}A}\,\mathsf{G}I^4}{b:\mathsf{X}\mathsf{G}A}\,\mathsf{X}I^3}{\dfrac{b:A\wedge\mathsf{X}\mathsf{G}A}{b:\mathsf{G}A\supset(A\wedge\mathsf{X}\mathsf{G}A)}\,\supset I^1}\,\wedge I$$

[7] This is not a problem of our formulation: all the finitary deduction systems for temporal logics equipped with at least the operators X and G have such a defect; see, e.g., [15, Ch. 6].

$(A6)$

$$
\begin{prooftree}
\AxiomC{$[b : A]^2$}
\AxiomC{$[b \leqslant c]^3$}
\AxiomC{$[b : \mathsf{G}(A \supset \mathsf{X}A)]^1$}
\AxiomC{$[b \leqslant b_i]^4$}
\RightLabel{$\mathsf{G}E$}
\BinaryInfC{$bb_i : A \supset \mathsf{X}A$}
\AxiomC{$[b_i : A]^4$}
\RightLabel{last}
\UnaryInfC{$bb_i : A$}
\RightLabel{$\supset E$}
\BinaryInfC{$bb_i : \mathsf{X}A$}
\AxiomC{$[b_i \lhd b_j]^4$}
\RightLabel{$\mathsf{X}E$}
\BinaryInfC{$bb_ib_j : A$}
\RightLabel{last}
\UnaryInfC{$b_j : A$}
\RightLabel{ind^4}
\TrinaryInfC{$c : A$}
\RightLabel{last}
\UnaryInfC{$bc : A$}
\RightLabel{$\supset^3$}
\UnaryInfC{$b : \mathsf{G}A$}
\RightLabel{$\supset I^2$}
\UnaryInfC{$b : A \supset \mathsf{G}A$}
\RightLabel{$\supset I^1$}
\UnaryInfC{$b : \mathsf{G}(A \supset \mathsf{X}A) \supset (A \supset \mathsf{G}A)$}
\end{prooftree}
$$

□

5 Conclusions

The introduction of the operator ∇ has allowed us to formalize the "history" of until and thus, via a proper translation, to give a labeled natural deduction system for a linear time logic endowed with ∇ that is also sound and complete with respect to LTL with until. As we remarked above, we see this work as spawning several different directions for future research. First, the "recipe" for dealing with until that we gave here is abstract and general, and thus provides the basis for formalizing deduction systems for temporal logics endowed with U, both linear and branching time. We are currently considering CTL^* and its sublogics as in [18,20] and are also working at a formal characterization of the class of logics that can be captured with our approach.

Second, the well-behaved nature of our approach, where each connective and operator has one introduction and one elimination rule, paves the way to a proof-theoretical analysis of the resulting natural deduction systems, e.g., to show proof normalization and other useful meta-theoretical properties, which we are tackling in current work. Moreover, we are also considering different optimizations of the rules. In particular, along the lines of the discussion about the rule *last* (and Corollary 3.6 and Definition 4.2), we are investigating to what extent we can use sequences as labels only when they are really needed, which would also simplify the proofs of normalization and other meta-properties. [8]

This is closely related to the formalization of the relationship between the class of LTL^l-formulas and that of LTL-formulas, which in turn will allow us to reason about the completeness of $\mathcal{N}(LTL_\nabla)$ with respect to the semantics of LTL_∇ and also to provide an axiomatization of LTL_∇ (thus treating it as a full-fledged logic as opposed to as a "service" logic for LTL as we did here).

In [16], an extension of a linear-time temporal logic with past is presented, where a unary operator *now* is used in order to fix a point of evaluation. When used in

[8] As an interesting side-track, we believe that the restrictions we imposed on formulas for the rule *last*, i.e., considering A^l and $A^{l\nabla}$, is closely related, at least in spirit, to the focus on *persistent* formulas when combining intuitionistic and classical logic so as to avoid the collapse of the two logics into one, see [6] but also [4,9]. We are, after all, considering here formulas stemming from two classes (if not two logics altogether), and it makes thus sense that they require different labeling (single instants and sequences).

combination with past operators, *now* allows to "forget" part of the past. The resulting logic is proved to be equally expressive to, but more succinct than, *LTL* with past.

Finally, it is worth observing that several works have considered *interval temporal logics*, e.g., [3,5,11,14,21]. While these works consider intervals explicitly, we have used them somehow implicitly here, as a means to formalize the dual nature of until via the history ∇, and this is another reason why it is interesting to reduce the use of label sequences as much as possible. A more detailed comparison of our approach with these works is left for future work.

Acknowledgement

This work was partially supported by the PRIN projects "CONCERTO" and "SOFT".

References

[1] Basin, D., C. Caleiro, J. Ramos and L. Viganò, *Labeled Tableaux for Distributed Temporal Logic*, J. Logic and Computation (2009, doi: 10.1093/logcom/exp022).

[2] Bolotov, A., O. Grigoriev and V. Shangin, *Automated natural deduction for propositional linear-time temporal logic.*, in: *Proc. TIME'07* (2007), pp. 47–58.

[3] Bowman, H. and S. Thompson, *A decision procedure and complete axiomatization of finite interval temporal logic with projection*, J. Logic and Computation **13** (2003).

[4] Caleiro, C. and J. Ramos, *Combining classical and intuitionistic implications*, in: *Proc. FroCoS'07*, LNCS 4720, 2007, pp. 118–132.

[5] Cerrito, S. and M. Cialdea Mayer, *Labelled Tableaux for Linear Time Temporal Logic over Finite Time Frames*, in: *Labelled Deduction*, Kluwer, 2000, pp. 130–144.

[6] Fariñas Del Cerro, L. and A. Herzig, *Combining classical and intuitionistic logic*, in: *Proc. FroCos'96*, 1996, pp. 93–102.

[7] Fisher, M., D. M. Gabbay and L. Vila, editors, "Handbook of Temporal Reasoning in Artificial Intelligence I," Elsevier, 2005.

[8] Gabbay, D. M., "Labelled Deductive Systems," Clarendon Press, 1996.

[9] Gabbay, D. M., *An overview of fibred semantics and the combination of logics*, in: *Proc. FroCoS'07*, LNCS 4720, 2007, pp. 1–56.

[10] Goldblatt, R. I., "Logics of Time and Computation," CSLI Lecture Notes, 1987.

[11] Goranko, V., A. Montanari, P. Sala and G. Sciavicco, *A general tableau method for propositional interval temporal logics: Theory and implementation*, J. Applied Logic **4** (2006), pp. 305–330.

[12] Gore, R., *Tableau methods for modal and temporal logics*, in: *Handbook of Tableau Methods*, Kluwer, 1999.

[13] Gough, G. D., *Decision procedures for temporal logic*, Technical Report UMCS-89-10-1, Department of Computer Science, University of Manchester (1984).

[14] Halpern, J. and Y. Shoham, *A propositional modal logic of time intervals*, JACM **38** (1991), pp. 935–962.

[15] Kröger, F., "Temporal logic of programs," Springer-Verlag, 1987.

[16] Laroussinie, N. Markey and P. Schnoebelen, *Temporal Logic with Forgettable Past*, in: *Proc. LICS'02*, pp. 383–392, 2002.

[17] Masini, A., L. Viganò and M. Volpe, *A History of Until* (Extended Version), arXiv:0910.4500, http://arxiv.org/abs/0910.4500, 2009.

[18] Masini, A., L. Viganò and M. Volpe, *A labeled natural deduction system for a fragment of CTL**, in: *Proc. LFCS'09*, LNCS 5407 (2009), pp. 338–353.

[19] Prawitz, D., "Natural Deduction, "Almquist and Wiskell, 1965.

[20] Reynolds, M., *A tableau for bundled CTL**, J. Logic and Computation **17** (2007).

[21] Schmitt, P. H. and J. Goubault-Larrecq, *A Tableau System for Linear-TIME Temporal Logic*, in: *Proc. TACAS'97*, LNCS 1217 (1997), pp. 130–144, See also the unpublished manuscript *A Tableau System for Full Linear Temporal Logic*.

[22] Schwendimann, S., *A New One-Pass Tableau Calculus for PLTL*, in: *Proc. Tableaux'98*, LNAI 1397 (1998), pp. 277–291.

[23] Simpson, A., "The Proof Theory and Semantics of Intuitionistic Modal Logic," Ph.D. thesis, School of Informatics, University of Edinburgh (1994).

[24] Viganò, L., "Labelled Non-Classical Logics," Kluwer, 2000.

[25] Viganò, L. and M. Volpe, *Labeled Natural Deduction Systems for a Family of Tense Logics*, in: *Proc. TIME'08* (2008), pp. 118–126.