Completeness for Logics on Finite Traces

Eric Campbell[1] and Michael Greenberg[2]

[1] Cornell University
ehc86@cornell.edu
[2] Pomona College
michael@cs.pomona.edu

Abstract. Temporal logics over finite traces are not the same as temporal logics over potentially infinite traces. We propose that existing methods for proving deductive completeness for infinite-trace logics are effective on their finite counterparts. To adapt proofs for infinite-trace logics, we "inject" finiteness: that is, we alter the proof structure to ensure that models are finite. As evidence for this claim, we offer deductive completeness results for two finite temporal logics: linear temporal logic over finite traces (LTL_f) and linear dynamic logic over finite traces (LDL_f). Both proofs of completeness follow a conventional, graph based, least fixed point structure. Roşu first proved completeness for LTL_f with a novel coinductive axiom; our proof uses fewer and more conventional axioms [12]. The proof for LDL_f is novel; it largely follows our LTL_f proof, using Brzozowski derivatives to define a transition function.

Keywords: finite temporal logic, linear temporal logic, linear dynamic logic, soundness, completeness, Brzozowski derivative

1 Introduction

Temporal logics have proven useful in a remarkable number of applications, in particular reasoning about reactive systems. To accommodate the nonterminating nature of such systems, temporal logics have used a possibly infinite model of time. For nearly thirty years after Pnueli's seminal work [11], the prevailing wisdom held that temporal logics could be 'truncated' to work with finite time. Researchers have recently overturned that conventional wisdom: some formulae are valid only in finite models [1, 5, 4].

Having realized that finite temporal logics differ from (possibly) infinite ones, we may wonder: how do these finite temporal logics behave? What of their model and proof theories? Can we adapt existing metatheoretical techniques from infinite settings, or must we come up with new ones? Reworking the model theory of temporal logics for finite time is an easy enough exercise: the standard model is a (possibly infinite) sequence of valuations on primitive propositions; to consider only finite models, simply restrict yourself to finite sequences of valuations. The proof theory is more challenging. In practice, it seems sufficient to (a) add an axiom indicating that the end of time comes, (b) add an axiom to say what happens when the end of time arrives, and (c) to relax (or strengthen)

axioms from the infinite logic that may not hold in finite settings. For an example of (c), consider LTL_f. It normally holds that the next modality commutes with implication, i.e., $\circ(\phi \Rightarrow \psi) \Leftrightarrow (\circ\,\phi \Rightarrow \circ\,\psi)$; in a finite setting, we must relax the if-and-only-if to merely the left-to-right direction.

Once we settle on a set of axioms, what does a proof of deductive completeness look like? We believe that it is possible to adapt existing techniques for infinite temporal logics to finite ones *directly*. As evidence, we offer two proofs of completeness: one for linear temporal logic over finite traces (LTL_f) and one for linear dynamic logic over finite traces (LDL_f). Both proofs have a conventional structure: we define a graph of *positive-negative pairs* of formulae (PNPs), following Kröger and Merz's presentation [9]. The only change we make to their construction is that when we prove our satisfiability lemma—the core property relating the PNP graph to provability—we "inject" finiteness, adding a formula guaranteeing a finite model.

We claim the following contributions:

- Evidence for the claim that the metatheory for infinite temporal logics readily adapts to finite temporal logics by means of *injecting finiteness* (Section 2 situates our work; Section 3 explains our model of finite time).
- A proof of deductive completeness for linear temporal logic on finite traces (LTL_f; Section 4) with fewer axioms than any prior proof [12].
- A novel proof of deductive completeness for linear dynamic logic on finite traces (LDL_f; Section 5).

2 Related work

Pnueli [11] proved his temporal logic programs to be sound and complete over traces of "discrete systems" which may or not be finite; Lichtenstein et al. [10] extended LTL with past-time operators and allowed more explicitly for the possibility of finite or infinite traces. Fischer and Ladner first devised propositional dynamic logic [8]; Vardi later introduced the linear variant, LDL [13].

Baier and McIlraith were the first to observe that some formulae are only valid in infinite models, and so LTL_f and other 'truncated' finite temporal logics differ from their infinite originals [1]. De Giacomo and Vardi showed that satisfiability and validity were PSAPCE-complete for these finite, relating LTL_f and LDL_f to other logics (potentially infinite LTL, FO[$<$], star-free regular expressions, MSO on finite traces) [5]; later, de Giacomo et al. were able to directly characterize when LTL_f and LDL_f formulae are sensitive to infiniteness [4]. De Giacomo and Vardi have also studied the synthesis problem for our two logics of interest [6, 7]. Most recently, D'Antoni and Veanes offered a decision procedure for MSO on finite sequences, but without a deductive completeness result [3].

Roşu [12] was the first to show a deductive completeness result for a finite temporal logic: he showed LTL_f is deductively complete by replacing the induction axiom with a *coinduction* axiom coInd: if $\vdash \bullet \Box\,\phi \Rightarrow \phi$ then $\vdash \phi$.[3] He shows

[3] In Roşu's paper, empty circles mean "weak next" and filled ones mean "next", while we follow Kröger and Merz and do the reverse [9].

that coInd is equivalent to the combination of the conventional induction axiom Ind (if $\vdash (\phi \Rightarrow \bullet\phi)$ then $\vdash \phi \Rightarrow \Box\phi$) axiom and a finiteness axiom Fin, $\Diamond \bullet \bot$.

Our goal is to show that existing, conventional methods for infinite temporal logics suffice for proving that finite temporal logics are deductively complete. For LTL_f, we take the conventional inductive framing, extending Kröger and Merz's axioms with the axiom Fin : $\Diamond \bullet \bot$, i.e., $\Diamond \mathsf{end}$ (we call this axiom Finite). Surprisingly, we are able to prove completeness with only six temporal axioms—one fewer than Roşu's seven, though he conjectures his set is minimal. It turns out that some of his axioms are in fact consequences of others—his $\mathrm{N}_\Box$ can be proved from $\mathrm{N}_\bullet$ and coInd (via Ind). Our results for LTL_f show that a smaller axiom set exists. In fact, we could go still smaller: using Roşu's proofs, we can replace Finite and Induction with coInd... only five axioms! We see our proof as offering a separate contribution, beyond shrinking the number of axioms needed. We follow Kröger and Merz's least fixed-point construction quite closely, adapting their proof from LTL to LTL_f by *injecting finiteness*. Our key idea is that existing techniques for infinite systems readily adapt to finite ones: we can reuse model theory which uses potentially infinite models so long as we can force the theory to work exclusively with finite models.

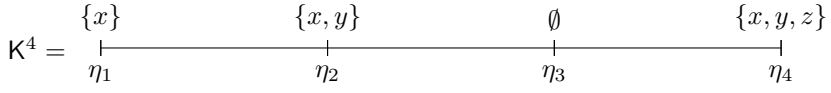
3 Modeling finite time

Both of our logics, LTL_f and LDL_f, share a model of finite time: *traces*. A trace over a fixed set of propositional variables is a (possibly infinite) sequence $(\eta_1, \eta_2, \dots)$ where η_i is a *valuation*, i.e., a function establishing the truth value ($\mathfrak{t}$ or $\mathfrak{f}$) for each propositional variable. We refer to each valuation as a 'state', with the intuition that each valuation represents a discrete moment in time. Formally:

Definition 1 (Valuations and Kripke structures). *Given a set of propositional variables* Var*, a* valuation *is a function* $\eta : \mathsf{Var} \to \{\mathfrak{t}, \mathfrak{f}\}$*. A* Kripke structure *or a* trace *is a finite, non-empty sequence of valuations; we write* $\mathsf{K}^n \in \mathsf{Model}_n$ *to refer to a model with* n *valuations, i.e.,* $\mathsf{K}^n = (\eta_1, \dots, \eta_n)$*.*

We particularly emphasize the finiteness of our Kripke structures, writing K^n and explicitly stating the number of valuations as a superscript each time. The number n is not directly accessible in either of our logics, though the size of models is observable in both of them (e.g., the LTL_f formula $\circ\circ\circ\top$ will be satisfiable only in models with four or more steps). Our traces are not only finite, but they are necessarily non-empty—both LTL_f and LDL_f would be trivial in empty models.

Suppose we have $\mathsf{Var} = \{x, y, z\}$. As a first example, the smallest possible model will be one with only one time step, $\mathsf{K}^1 = \eta_1$, where η_1 is a function from Var to the booleans, i.e., a subset of Var. As a more complex example, consider the following model K^4 with four time steps:



Both of our logics use Kripke structures to interpret formulae, defining a function $\mathsf{K}_i^n : \mathsf{Formula} \to \{\mathfrak{t}, \mathfrak{f}\}$. (Put another way: we define a function $\mathsf{interp} : \mathsf{Model}_n \times \{1, \ldots, n\} \times \mathsf{Formula} \to \{\mathfrak{t}, \mathfrak{f}\}$.) We lift this interpretation function to define validity and satisfiability abstractly for both logics.

Definition 2 (Satisfiability and validity in a Kripke structure). *Given an interpretation function* $\mathsf{K}_i^n : \mathsf{Formula} \to \{\mathfrak{t}, \mathfrak{f}\}$*, we say for* $\phi \in \mathsf{Formula}$*:*

- ϕ *is* satisfiable in K^n *iff* $\mathsf{K}_1^n(\phi) = \mathfrak{t}$*;*
- ϕ *is* satisfiable *iff* $\exists \mathsf{K}^n$ *such that* ϕ *is satisfiable in* K^n*;*
- $\mathsf{K}^n \models \phi$ *(pronounced "*K^n *models* ϕ*") iff* $\forall 1 \leq i \leq n,\ \mathsf{K}_i^n(\phi) = \mathfrak{t}$*;*
- $\models \phi$ *(pronounced "*ϕ *is valid") iff* $\forall \mathsf{K}^n,\ \mathsf{K}^n \models \phi$*; and*
- $\mathcal{F} \models \phi$ *for* $\mathcal{F} \subseteq \mathsf{Formula}$ *(pronounced "*ϕ *is valid under* $\mathcal{F}$*") iff* $\forall \mathsf{K}^n$*, if* $\forall \psi \in \mathcal{F},\ \mathsf{K}^n \models \psi$ *then* $\mathsf{K}^n \models \phi$*.*

4 LTL_f: linear temporal logic on finite traces

Linear temporal logic is a logic for reasoning on potentially infinite traces. The syntax of linear temporal logic on finite traces (LTL_f) is identical to that of its (potentially) infinite counterpart. We define LTL_f as propositional logic with two temporal operators (Figure 1). The propositional fragment comprises: variables v from some fixed set of propositional variables Var; the false proposition $\bot$; and implication $\phi \Rightarrow \psi$. The temporal fragment comprises two operators: the *next modality*, written $\circ\phi$; and, *weak until*, written $\phi \mathcal{W} \psi$.

Given these parts, we can encode a more conventional looking logic, with the usual logical operators and an enriched set of temporal operators. Of these standard encodings, we remark on two in particular: end, the end of time, and $\bullet\phi$, the *weak next* modality. In the usual (potentially infinite) semantics, it is generally the case that $\circ\top$ holds, i.e., that the true proposition holds in the next state, i.e., that there *is* a next state. But at the end of time, there is no next state—and so $\circ\top$ ought not adhere. In every state *but* the last, we have $\circ\top$ as usual. We can therefore define $\mathsf{end} = \neg\circ\top$—if end holds, then we must be at the end of time. We define the *weak next* modality as $\bullet\phi = \neg\circ\neg\phi$. In an infinite model, next and weak next generally coincide. But we have $\bullet\top$ in *every* state, but $\circ\top$ in all but the last. That is, weak next is insensitive to the end of time—when $\bullet\phi$ holds for all ϕ—but $\circ\phi$ is senstitive to the end of time—when $\circ\phi$ fails for all ϕ. To realize these intuitions, we must define our model.

The simple, standard model for LTL is a *trace*; we will use finite traces (Definition 1). Given a Kripke structure K^n, we assign a truth value to a proposition ϕ at time step $1 \leq i \leq n$ with the function $\mathsf{K}_i^n(\phi)$, defined as a fixpoint on formulae. The definitions for K_i^n in the propositional fragment are straightforward implementations of the conventional operations. The definitions for K_i^n in the temporal fragment also assign the usual meanings, being mindful of the end of time. When there is no next state, the formula $\circ\phi$ is necessarily false; when there is no next state, the formula $\phi \mathcal{W} \psi$ degenerates into $\phi \vee \psi$. Why? Suppose we are at the end of time; one of two cases adheres. Either (weakly) we have ϕ until

Syntax

$$\begin{array}{ll} \text{Variables} & v \in \mathsf{Var} \\ \text{LTL}_f \text{ formulae} & \phi, \psi \in \text{LTL}_f ::= v \mid \bot \mid \phi \Rightarrow \psi \mid \circ \phi \mid \phi \,\mathcal{W}\, \psi \end{array}$$

Encodings

$$\neg\phi = \phi \Rightarrow \bot \qquad \top = \neg\bot \qquad \phi \vee \psi = \neg\phi \Rightarrow \psi \qquad \phi \wedge \psi = \neg(\phi \vee \psi)$$
$$\mathsf{end} = \neg \circ \top \qquad \bullet \phi = \neg \circ \neg\phi \qquad \Box \phi = \phi \,\mathcal{W}\, \bot \qquad \Diamond \phi = \neg \Box \neg\phi$$
$$\phi \,\mathcal{U}\, \psi = \phi \,\mathcal{W}\, \psi \wedge \Diamond \psi$$

Semantics $\boxed{\mathsf{K}_i^n : \text{LTL}_f \to \{\mathfrak{t}, \mathfrak{f}\}}$

$$\mathsf{K}_i^n(v) = \eta_i(v)$$
$$\mathsf{K}_i^n(\bot) = \mathfrak{f}$$
$$\mathsf{K}_i^n(\phi \Rightarrow \psi) = \begin{cases} \mathfrak{t} & \mathsf{K}_i^n(\phi) = \mathfrak{f} \text{ or } \mathsf{K}_i^n(\psi) = \mathfrak{t} \\ \mathfrak{f} & \text{otherwise} \end{cases}$$
$$\mathsf{K}_i^n(\circ \phi) = \begin{cases} \mathsf{K}_{i+1}^n(\phi) & i < n \\ \mathfrak{f} & i = n \end{cases}$$
$$\mathsf{K}_i^n(\phi \,\mathcal{W}\, \psi) = \begin{cases} \mathfrak{t} & \forall i \le j \le n,\ \mathsf{K}_j^n(\phi) = \mathfrak{t} \text{ or} \\ & \exists i \le k \le n,\ \mathsf{K}_k^n(\psi) = \mathfrak{t} \text{ and } \forall i \le j < k,\ \mathsf{K}_j^n(\phi) = \mathfrak{t} \\ \mathfrak{f} & \text{otherwise} \end{cases}$$

Proof theory $\boxed{\vdash \subseteq 2^{\text{LTL}_f} \times \text{LTL}_f}$

Axioms

all propositional tautologies — TAUT

$\vdash \bullet(\phi \Rightarrow \psi) \Leftrightarrow (\bullet\phi \Rightarrow \bullet\psi)$ — WKNEXTDISTR

$\vdash \mathsf{end} \Rightarrow \neg \circ \phi$ — ENDNEXTCONTRA

$\vdash \Diamond \mathsf{end}$ — FINITE

$\vdash \phi \,\mathcal{W}\, \psi \Leftrightarrow \psi \vee (\phi \wedge \bullet(\phi \,\mathcal{W}\, \psi))$ — WKUNTILUNROLL

$$\frac{\vdash \phi}{\vdash \bullet\phi}$$ — WKNEXTSTEP

$$\frac{\vdash \phi \Rightarrow \psi \quad \vdash \phi \Rightarrow \bullet\phi}{\vdash \phi \Rightarrow \Box\psi}$$ — INDUCTION

Consequences

$\vdash \neg(\circ \top \wedge \circ \bot)$ — NEXTCONTRA

$\vdash \neg \circ \phi \Leftrightarrow \mathsf{end} \vee \circ \neg\phi$ — NEXTNEG

$\vdash \bullet\phi \Leftrightarrow \circ\phi \vee \mathsf{end}$ — NEXTWKNEXT

$\vdash \bullet(\phi \wedge \psi) \Leftrightarrow \bullet\phi \wedge \bullet\psi$ — WKNEXTCONJ

$\vdash \neg\bullet\phi \Rightarrow \bullet\neg\phi$ — WKNEXTNEG

$\vdash \Box\phi \Leftrightarrow \phi \wedge \bullet\Box\phi$ — ALWAYSUNROLL

$\mathcal{F} \vdash \phi$ iff assuming $\vdash \psi$ for each $\psi \in \mathcal{F}$ we have $\vdash \phi$

Fig. 1. LTL_f syntax, semantics, and proof theory

the end of time (now!), or we have ψ and have satisfied the until. We can verify our earlier intuitions about end and $\bullet\,\phi$. Observe that $\mathsf{K}_i^n(\mathsf{end}) = \mathsf{t}$ exactly when $i = n$; similarly, $\mathsf{K}_i^n(\bullet\,\top) = \mathsf{t}$ for all $1 \leq i \leq n$.

By way of example, consider K^4 from Section 3. We have $\mathsf{K}^4 \models y \Rightarrow x$, because $\mathsf{K}_i^4(y \Rightarrow x) = \mathsf{t}$ for all i, i.e., whenever y holds, so does x. Similarly, $x \,\mathcal{W}\, y$ is satisfiable in K^4 with $k = 2$; $z \,\mathcal{W}\, x$ is also satisfiable in K^4, but trivially with $k = 1$. The formula $\Box\, z$ is not satisfiable in K^4, but $\mathsf{K}^4 \models \Diamond\, z$. We lift K_i^n to satisfiability and validity in the usual way (Definition 2).

For our axioms, we adapt Kröger and Merz's presentation [9]. Two axioms are new: Finite says that time will eventually end; EndNextContra says that at the end of time, there is no next state. Other axioms are lightly adapted: wherever we would ordinarily use the next modality, $\circ\,\phi$, we instead use weak next, $\bullet\,\phi$. The axioms with strong next are unsound in finite models. We can, however, characterize the relationship between the next modality, negation, and the end of time ("Consequences" in Figure 1).

Roşu proves completeness with a slightly different set of axioms, replacing Finite and Induction with a single *coinduction* axiom he calls coInd:

$$\frac{\vdash \bullet\,\Box\,\phi \Rightarrow \phi}{\vdash \phi}$$

He proves that coInd is equivalent to the conjunction of Finite and Induction, so it does not particularly matter which axioms we choose. In order to emphasize how little must change to make our logic finite, we keep our presentation as close to Kröger and Merz's as possible[4].

Proving that our axioms are sound is, as usual, relatively straightforward.

Theorem 3 ($\mathbf{LTL}_f$ soundness). *If $\vdash \phi$ then $\models \phi$.*

Proof. By induction on the derivation of $\vdash \phi$.

4.1 Completeness

To show deductive completeness for LTL_f, we must find that if $\models \phi$ then $\vdash \phi$. To do so we will construct a graph that does two things at once: first, paths from the root of the graph to a terminal state correspond to Kripke structures which ϕ satisfies; second, consistency properties in the graph relate to the provability of the underlying formula ϕ.

Our construction follows the standard least-fixed point approach found in Kröger and Merz's book [9]: we construct a graph whose nodes assign truth values to each subformula of our formula of interest, ϕ, by putting each subformula in either the true, "positive" set or in the false, "negative" set. Our completeness proof ultimately construct a graph for the *negation* of ϕ, showing that the negated graph has no satisfiable models—therefore showing that $\vdash \neg\neg\phi$

[4] They use a less-expressive syntax, omitting $\mathcal{W}$ and $\mathcal{U}$. We extend their methodology to include these operators.

is unprovable, and so $\vdash \phi$ is provable (since the propositional logic undergirding LTL_f is classical).

What about the 'f' in LTL_f? Nothing described so far differs in any way from the Henkin-Hasenjaeger graph approach used by Kröger and Merz [9]. Kröger and Merz's graphs were always finite, but their notion of satisfying paths allows for infinite paths. We restrict our attention to terminating paths: paths where not only is our formula of interest satisfied, but so is $\Diamond\,\mathsf{end}$. To ensure such paths exist, we *inject* $\Diamond\,\mathsf{end}$ when we create the graph.

The proof follows the following structure: we define the nodes of the graph (Definition 4); we define the edge relation on the graph (Figure 2) and show that it maps appropriately to time steps in the proof theory (Lemma 6 finds a consistent successor; Lemma 5 shows the successor is a state in our graph); we show that the graph structure results in a finite structure with appropriate consistency properties (Lemma 10); we define which paths in the graph represent our Kripke structure of interest (Lemma 12 shows that our graph's transitions correspond to the semantics; Lemma 14 guarantees that we have appropriate finite models). The final proof comes in two parts: we show that consistent graphs correspond to satisfiable formulae (Theorem 15), which is enough to show completeness (Theorem 16).

Definition 4 (Positive-negative pairs (PNPs)). *A* positive-negative pair *(PNP) $\mathcal{P}$ is a pair of sets of formulae* $(\mathsf{pos}(\mathcal{P}),\ \mathsf{neg}(\mathcal{P}))$*. We refer to the collected formulas of $\mathcal{P}$ as $\mathcal{F}_{\mathcal{P}} = \mathsf{pos}(\mathcal{P}) \cup \mathsf{neg}(\mathcal{P})$; we call the set of all PNPs* PNP*.*

We write the literal interpretation *of $\mathcal{P}$ as $\widehat{\mathcal{P}} = \bigwedge_{\phi \in \mathsf{pos}(\mathcal{P})} \phi \wedge \bigwedge_{\psi \in \mathsf{neg}(\mathcal{P})} \psi$.*

We say $\mathcal{P}$ is inconsistent *if $\vdash \neg\widehat{\mathcal{P}}$; conversely, $\mathcal{P}$ is* consistent *when it is not the case that $\vdash \neg\widehat{\mathcal{P}}$.*

We write $\mathcal{P} \preceq \mathcal{Q}$ (read "$\mathcal{P}$ is extended by $\mathcal{Q}$" or "$\mathcal{Q}$ extends $\mathcal{P}$") when $\mathcal{Q}$'s positive and negative sets subsume $\mathcal{P}$'s (Figure 2). We say $\mathcal{P}$ is complete when $\mathcal{F}_{\mathcal{P}} = \tau(\mathcal{P})$. We say a complete PNP $\mathcal{Q}$ is a *completion* of $\mathcal{P}$ when $\mathcal{P} \preceq \mathcal{Q}$ and $\mathcal{Q}$ is consistent. We define the set of all consistent completions of a given PNP $\mathcal{P}$ with $\mathsf{comps}(\mathcal{P})$. Our goal is to generate successors states to build a graph of PNPs; the step function σ takes a PNP and generates those formulae which must hold in the next step, thereby characterizing the transitions in our graph.

Lemma 5 (Transitions are provable). *For all PNPs $\mathcal{P}$, we have $\vdash \widehat{\mathcal{P}} \Rightarrow \bullet\widehat{\sigma(\mathcal{P})}$.*

Proof. *By cases on the clauses of σ.*

Lemma 6 (Consistent completions are provable). *For all consistent PNPs $\mathcal{P}$, we have $\vdash \widehat{\mathcal{P}} \Rightarrow \bigvee_{\mathcal{Q} \in \mathsf{comps}(\mathcal{P})} \widehat{\mathcal{Q}}$.*

Proof. *By showing that $\vdash \bigvee_{\mathcal{P} \in \mathsf{assigns}(\mathcal{F})} \widehat{\mathcal{P}}$ (by induction on the size of $\mathcal{F}$) and that for all consistent PNPs $\mathcal{P}$ and for all $\mathcal{Q} \in \mathsf{assigns}(\widehat{\mathcal{P}})$, if $\vdash \widehat{\mathcal{P}} \Rightarrow \widehat{\mathcal{Q}}$ then $\mathcal{Q} \in \mathsf{comps}(\mathcal{P})$.*

Transition functions $\boxed{\sigma_i^\bullet : \mathsf{PNP} \to 2^{\mathrm{LTL}_f}}$ $\boxed{\sigma : \mathsf{PNP} \to \mathsf{PNP}}$

$$\sigma_1^+(\mathcal{P}) = \{\phi \mid \circ\phi \in \mathsf{pos}(\mathcal{P})\}$$
$$\sigma_2^+(\mathcal{P}) = \{\phi \mathcal{W} \psi \mid \phi \mathcal{W} \psi \in \mathsf{pos}(\mathcal{P}), \psi \in \mathsf{neg}(\mathcal{P})\}$$
$$\sigma_3^-(\mathcal{P}) = \{\phi \mid \circ\phi \in \mathsf{neg}(\mathcal{P})\}$$
$$\sigma_4^-(\mathcal{P}) = \{\phi \mathcal{W} \psi \mid \phi \mathcal{W} \psi \in \mathsf{neg}(\mathcal{P}),\ \phi \in \mathsf{pos}(\mathcal{P})\}$$
$$\sigma(\mathcal{P}) = (\sigma_1^+(\mathcal{P}) \cup \sigma_2^+(\mathcal{P}),\ \sigma_3^-(\mathcal{P}) \cup \sigma_4^-(\mathcal{P}))$$

Closure $\boxed{\tau : \mathrm{LTL}_f \to 2^{\mathrm{LTL}_f}}$ $\boxed{\tau : 2_f^{\mathrm{LTL}} \to 2^{\mathrm{LTL}_f}}$ $\boxed{\tau : \mathsf{PNP} \to 2^{\mathrm{LTL}_f}}$

$$\tau(v) = \{v\} \qquad \tau(\bot) = \{\bot\}$$
$$\tau(\phi \Rightarrow \psi) = \{\phi \Rightarrow \psi\} \cup \tau(\phi) \cup \tau(\psi) \qquad \tau(\circ\phi) = \{\circ\phi\}$$
$$\tau(\phi \mathcal{W} \psi) = \{\phi \mathcal{W} \psi\} \cup \tau(\phi) \cup \tau(\psi)$$
$$\tau(\mathcal{F}) = \textstyle\bigcup_{\phi \in \mathcal{F}} \tau(\phi) \qquad \tau(\mathcal{P}) = \tau(\mathcal{F}_\mathcal{P})$$

Extensions, completions, and possible assignments $\boxed{\preceq \subseteq \mathsf{PNP} \times \mathsf{PNP}}$

$\boxed{\mathsf{comps} : 2^{\mathrm{LTL}_f} \to 2^{\mathsf{PNP}}}$ $\boxed{\mathsf{comps} : \mathsf{PNP} \to 2^{\mathsf{PNP}}}$ $\boxed{\mathsf{assigns} : 2^{\mathrm{LTL}_f} \to 2^{\mathsf{PNP}}}$

$$\mathcal{P} \preceq \mathcal{Q} \text{ iff } \mathsf{pos}(\mathcal{P}) \subseteq \mathsf{pos}(\mathcal{Q}) \text{ and } \mathsf{neg}(\mathcal{P}) \subseteq \mathsf{neg}(\mathcal{Q})$$
$$\mathsf{assigns}(\mathcal{F}) = \{\mathcal{P} \mid \mathcal{F}_\mathcal{P} = \tau(\mathcal{F})\} \quad \mathsf{comps}(\mathcal{P}) = \{\mathcal{Q} \mid \mathcal{F}_\mathcal{Q} = \tau(\mathcal{P}),\ \mathcal{P} \preceq \mathcal{Q},\ \mathcal{Q} \text{ consistent}\}$$

Fig. 2. Step and closure functions; extensions and completions

Having established the fundamental properties of consistent completions, we set about defining the structure on which we build our proof. We show that, starting from a PNP formed from a given formula, we can construct a graph where nodes are PNPs and a node $\mathcal{P}$'s successors are consistent completions of $\sigma(\mathcal{P})$.

Definition 7 (Proof graphs). *For a consistent and complete PNP $\mathcal{P}$ (i.e., it is not the case that $\vdash \neg\widehat{\mathcal{P}}$ and $\mathcal{F}_\mathcal{P} = \tau(\mathcal{P})$), we define a* proof graph *$\mathcal{G}_\mathcal{P}$ as follows: (a) $\mathcal{P}$ is the* root *of $\mathcal{G}_\mathcal{P}$; (b) $\mathcal{P}$ has an edge to the root of $\mathcal{G}_\mathcal{Q}$ for each $\mathcal{Q} \in \mathsf{comps}(\sigma(\mathcal{P}))$. The $\mathcal{Q} \in V(\mathcal{G}_\mathcal{P})$ are those PNPs reachable from $\mathcal{P}$.*

Is this graph finite?[5] To see that $\mathcal{G}_\mathcal{P}$ is indeed finite, observe that each node of $\mathcal{G}_\mathcal{P}$ is a complete PNP $\mathcal{Q}$ where $\mathcal{F}_\mathcal{Q} \subseteq \tau(\mathcal{P})$ (because both σ and comps are non-increasing). There are therefore a finite number of nodes in $\mathcal{G}_\mathcal{P}$.

The innovation in adapting the completeness proof to finite time is *finiteness injection*, where we make sure that $\Diamond\,\mathsf{end}$ is in the positive set of the root of

[5] In fact, Kröger and Merz [9] call this graph an "infinite tree" in their proof of completeness for potentially infinite LTL.

the proof graph we construct to show completeness. After injecting finiteness, every node of the proof graph will either have end in its positive set (and no successors) or all of its successors have $\Diamond\,\mathsf{end}$ in their positive set.

Lemma 8 (end injection is invariant). *If $\mathcal{P}$ is a consistent and complete PNP with $\Diamond\,\mathsf{end} \in \mathsf{pos}(\mathcal{P})$, then either:*

- $\mathsf{end} \in \mathsf{pos}(\mathcal{P})$ *and $\mathcal{P}$ has no successors (i.e., $\mathsf{comps}(\sigma(\mathcal{P})) = \emptyset$), or*
- $\mathsf{end} \in \mathsf{neg}(\mathcal{P})$ *and for all $\mathcal{Q} \in \mathsf{comps}(\sigma(\mathcal{P}))$, we have $\Diamond\,\mathsf{end} \in \mathsf{pos}(\mathcal{Q})$.*

Proof. *Recall that $\Diamond\,\mathsf{end}$ desugars to $\neg(\neg\neg \circ \top \;\mathcal{W}\; \bot)$. Since $\mathcal{P}$ is complete, we know that $\mathsf{end} \in \mathcal{F}_{\mathcal{P}}$. By cases on where end appears.*

We can go further, showing that $\Diamond\,\mathsf{end}$ is in fact in *every* node's positive set.

Lemma 9 (Nodes are consistent and complete). *For all consistent and complete PNPs $\mathcal{P}$, every node $\mathcal{Q} \in \mathcal{G}_{\mathcal{P}}$ is consistent and complete. If $\Diamond\,\mathsf{end} \in \mathsf{pos}(\mathcal{P})$, then $\Diamond\,\mathsf{end} \in \mathsf{pos}(\mathcal{Q})$.*

Proof. *By induction on the length of the shortest path from $\mathcal{P}$ to $\mathcal{Q}$ in $\mathcal{G}_{\mathcal{P}}$, using Lemma 8.*

Each node has the potential for successors: for each node $\mathcal{Q} \in \mathcal{G}_P$, we can prove that $\widehat{\mathcal{Q}}$ implies the disjunction of every other node's literal interpretation.

Lemma 10 (Step implication). *For all consistent and complete PNPs $\mathcal{P}$ where $\Diamond\,\mathsf{end} \in \mathsf{pos}(\mathcal{P})$ then $\vdash \bigvee_{\mathcal{Q}\in\mathcal{G}_{\mathcal{P}}} \widehat{\mathcal{Q}} \Rightarrow \bullet \bigvee_{\mathcal{Q}\in\mathcal{G}_{\mathcal{P}}} \widehat{\mathcal{Q}}$.*

Proof. *By Lemma 5, we know that $\vdash \widehat{\mathcal{Q}} \Rightarrow \bullet\,\widehat{\sigma(\mathcal{Q})}$.*

By Lemma 9, we know that $\mathcal{Q}$ is consistent and complete and $\Diamond\,\mathsf{end} \in \mathsf{pos}(\mathcal{Q})$. Since $\mathcal{Q}$ is complete, we know that $\mathsf{end} \in \mathcal{F}_{\mathcal{Q}}$. We go by cases on where end is.

We have so far established that the proof graph $\mathcal{G}_{\mathcal{P}}$ is rooted at $\mathcal{P}$, preserves any finiteness we may inject, and has provable successors. We are nearly done: we show that our proof graph corresponds to a Kripke structure which models $\mathcal{P}$.

Definition 11 (Terminal nodes and paths). *A node $\mathcal{Z} \in \mathcal{G}_{\mathcal{P}}$ is* terminal *when $\circ\top \in \mathsf{neg}(\mathcal{Z})$. A path $\mathcal{P}_1, \ldots, \mathcal{P}_n$ is* terminal *when $\mathcal{P}_n$ is terminal.*

Lemma 12 (Proof graphs match the semantic model). *For all consistent and complete PNPs $\mathcal{P}$, if $\mathcal{P}_1, \mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$ is a terminal path in $\mathcal{G}_{\mathcal{P}}$, then for all i:*

1. *For all formulae ϕ, if $\circ\,\phi \in \mathcal{F}_{\mathcal{P}_i}$ then $\circ\,\phi \in \mathsf{pos}(\mathcal{P}_i)$ iff $\phi \in \mathsf{pos}(\mathcal{P}_{i+1})$.*
2. *For all formulae ϕ and ψ, if $\phi\;\mathcal{W}\;\psi \in \mathcal{F}_{\mathcal{P}_i}$ then $\phi\;\mathcal{W}\;\psi \in \mathsf{pos}(\mathcal{P}_i)$ iff either $\phi \in \mathsf{pos}(P_j)$ for all $j \geq i$ or there is some $k \geq i$ such that $\psi \in \mathsf{pos}(\mathcal{P}_k)$ and $\forall i \leq j < k,\ \phi \in \mathsf{pos}(\mathcal{P}_j)$.*

Proof. *The first part follows by consistency of node (Lemma 9). The second part goes by induction on the length of the path.*

Here we slightly depart from Kröger and Merz's presentation: since their models can be infinite, they must make sure that their paths are able to in some sense 'fulfill' temporal predicates. We, on the other hand, know that all of our paths will be finite, so our reasoning can be simpler. First, there must exist some terminal node.

Lemma 13 (Injected finiteness guarantees terminal nodes). *For all consistent and complete PNPs $\mathcal{P}$, if $\Diamond\,\mathsf{end} \in \mathsf{pos}(\mathcal{P})$ then there is a terminal node $\mathcal{Z} \in \mathcal{G}_{\mathcal{P}}$.*

Proof. By Lemma 10.

Next, our proof graph is constructed to be connected, so the existence of a terminal node implies the existence of a terminal path.

Corollary 14 (Injected finiteness guarantees terminal paths). *For all consistent and complete PNPs $\mathcal{P}$, if $\Diamond\,\mathsf{end} \in \mathsf{pos}(\mathcal{P})$ then there is a terminal path $\mathcal{P}, \mathcal{P}_2, \ldots, \mathcal{P}_{n-1}, \mathcal{Z} \in \mathcal{G}_{\mathcal{P}}$.*

Proof. By Lemma 13, there exists some terminal node $\mathcal{Z} \in \mathcal{G}_{\mathcal{P}}$. Since $\mathcal{G}_{\mathcal{P}}$ is constructed by iterating comps *and σ on $\mathcal{P}$, there must exist some $\mathcal{P}_{n-1}$ such that $\mathcal{Z} \in \mathsf{comps}(\sigma(\mathcal{P}_{n-1}))$, and some $\mathcal{P}_{n-2}$ such that $\mathcal{P}_{n-1} \in \mathsf{comps}(\sigma(\mathcal{P}_{n-2}))$ and so on back to $\mathcal{P}$—yielding a path.*

We can now prove the key lemma: consistent PNPs are satisfiable—a proof graph for a consistent PNP $\mathcal{P}$ induces a Kripke structure modeling $\mathcal{P}$'s literal interpretation, $\widehat{\mathcal{P}}$.

Theorem 15 ($\mathbf{LTL}_f$ satisfiability). *If $\mathcal{P}$ is a consistent PNP, then $\widehat{\mathcal{P}}$ is satisfiable.*

Proof. Since $\mathcal{P}$ is consistent, so is adding $\Diamond\, END$ to its positive set. The proof graph with injected finiteness $\mathcal{G}_{\mathcal{P}}$ has a terminal path of length n for some n (Corollary 14); construct a Kripke structure of length n, where $\mathsf{pos}(\mathcal{P}_i)$ determines the ith valuation. By induction on n, $\mathsf{K}_i^n(\phi) = \mathfrak{t}$ iff $\phi \in \mathsf{pos}(\mathcal{P}_i)$ for all $\phi \in \mathcal{F}_{\mathcal{P}}$ by Lemma 12. Therefore $\mathsf{K}_1^n(\widehat{\mathcal{P}}) = \mathfrak{t}$.

Finally, we can show completeness. The proof is the usual one, where we to find a proof of ϕ we try to see if $\neg\phi$ is satisfiable—if not, then the PNP for $\neg\phi$ will be inconsistent... and so $\vdash \neg\neg\phi$, which yields $\vdash \phi$.

Theorem 16 ($\mathbf{LTL}_f$ completeness). *If $\models \phi$ then $\vdash \phi$.*

Proof. If $\models \phi$, then for all Kripke structures K^n, we have $\mathsf{K}_i^n(\phi) = \mathfrak{t}$ for all i. Conversely, it must also be the case that $\mathsf{K}_i^n(\neg\phi) = \mathfrak{f}$ for all i, and so $\neg\phi$ is unsatisfiable. In other words, the PNP $(\emptyset, \{\phi\})$ is unsatisfiable. By the contrapositive of Theorem 15, it must be the case that $(\emptyset, \{\phi\})$ is inconsistent, i.e., $\vdash \neg\neg\phi$. By Taut*, we can conclude that $\vdash \phi$.*

Syntax

$$
\begin{array}{ll}
\text{Variables} & v, a, b \in \mathsf{Var} \\
\text{LDL}_f \text{ programs} & \alpha, \beta, \gamma \in \mathsf{Program} ::= v \mid \alpha + \beta \mid \alpha; \beta \mid \alpha^* \\
\text{LDL}_f \text{ formulae} & \phi, \psi, \theta \in \mathsf{Formula} ::= v \mid \bot \mid \phi \vee \psi \mid \neg\phi \mid \langle\alpha\rangle\, \phi
\end{array}
$$

Encodings

$$
\mathsf{any} = \sum_{v \in \mathsf{Var}} v \qquad \mathsf{end} = \neg\langle \mathsf{any} \rangle \top \qquad [\alpha]\, \phi = \neg\, \langle\alpha\rangle\, \neg\phi
$$

Semantics $\boxed{\mathsf{K}_i^n : \mathsf{Formula} \to \{\mathsf{t}, \mathsf{f}\}}$ $\boxed{\mathcal{R}(\alpha, \mathsf{K}^n) \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}}$

$$
\begin{array}{rl}
\mathsf{K}_i^n(\bot) & = \mathsf{f} \\
\mathsf{K}_i^n(v) & = \eta_i(v) \\
\mathsf{K}_i^n(\phi \vee \psi) & = \mathsf{t} \text{ iff } \mathsf{K}_i^n(\phi) = \mathsf{t} \text{ or } \mathsf{K}_i^n(\psi) = \mathsf{t} \\
\mathsf{K}_i^n(\neg\phi) & = \mathsf{t} \text{ iff } \mathsf{K}_i^n(\phi) = \mathsf{f} \\
\mathsf{K}_i^n(\langle\alpha\rangle\, \phi) & = \mathsf{t} \text{ iff } \exists j \in [i, n] \text{ such that } (i, j) \in \mathcal{R}(\alpha, \mathsf{K}^n) \text{ and } \mathsf{K}_j^n(\phi) = \mathsf{t}
\end{array}
$$

$$
\begin{array}{rl}
\mathcal{R}(v, \mathsf{K}^n) & = \{(i, i+1) \mid \mathsf{K}_i^n(v) = \mathsf{t}\} \\
\mathcal{R}(\alpha + \beta, \mathsf{K}^n) & = \mathcal{R}(\alpha, \mathsf{K}^n) \cup \mathcal{R}(\beta, \mathsf{K}^n) \\
\mathcal{R}(\alpha; \beta, \mathsf{K}^n) & = \{(i, j) \mid \exists k \text{ such that } (i, k) \in \mathcal{R}(\alpha, \mathsf{K}^n) \text{ and } (k, j) \in \mathcal{R}(\beta, \mathsf{K}^n)\} \\
\mathcal{R}(\alpha^*, \mathsf{K}^n) & = \{(i, i)\} \cup \{(i, j) \mid \exists k \text{ such that } (i, k) \in \mathcal{R}(\alpha, \mathsf{K}^n) \\
& \qquad \text{and } (k, j) \in \mathcal{R}(\alpha^*, \mathsf{K}^n)\}
\end{array}
$$

Fig. 3. LDL_f syntax and semantics

5 LDL_f

Finite-time linear dynamic logic (LDL_f) adapts linear dynamic logic (LDL [13]), which itself adapts propositional dynamic logic [8]. All three of these logics share a syntax (Figure 3); the differences come in the model. The core idea is to allow formulae to reason explicitly about state transitions.

Concretely, LDL_f's syntax has two layers: *formulae* ϕ and *programs* (a/k/a path expressions) α. Formulae are the top-level construct, comprising propositional logic (here, variables $a \in \mathsf{Var}$, false ($\bot$), negation ($\neg$), and disjunction $\vee$)[6] and a modal *path operator* $\langle\alpha\rangle\, \phi$, where α is a program and ϕ is another formula. We pronounce this path operator literally as "angle α ϕ", meaning that "after we move through time along an α path, ϕ holds". Programs α are regular expressions: they include propositional variables a/k/a *primitive programs* $a \in \mathsf{Var}$, alternation $\alpha + \beta$, concatenation $\alpha; \beta$, and Kleene star α^*. We additionally encode three other forms: any, which represents any single primitive program; end, which represents the end of time, and $[\alpha]\, \phi$, which is the dual of $\langle\alpha\rangle\, \phi$.

[6] For LTL_f, we used $\bot$ and $\Rightarrow$ as our logical basis, but it is more convenient for our proofs to use this $\bot$, $\neg$, and $\vee$ for LDL_f.

Before defining the model, we offer some examples of formulae and their intuitive meanings. The formula $\phi = \langle \mathsf{any} \rangle\, y$ says that after one time step, y will hold; the formula $\psi = [z]\, y$ says that after every z-step, y will hold; the formula $\theta = \langle x + y \rangle\, x \wedge y$ says that after taking a step with either x or y, x and y will both hold; the formula $\chi = \neg \langle \mathsf{any}^* \rangle\, z$ says that it is not the case that after any number of time steps, z will hold; χ is an encoding of the LTL_f formula $\neg \Diamond z$ [5]. To better undstand these formulae, we define our semantics concretely.

LDL_f is distinct from LDL and PDL in its model. Models of PDL branch arbitrarily; models in LDL are linear; models in LDL_f are linear and finite. We reuse finite Kripke structures (Section 3), to define semantics (Figure 3), though we need two functions: one to assign meaning to formulaes, written $\mathsf{K}^n_i : \mathsf{Formula} \to \{\mathfrak{t}, \mathfrak{f}\}$ just like in LTL_f, and one to assign meaning to programs, written $\mathcal{R}(,) : \mathsf{Program} \times \mathsf{Model}_n \to 2^{\{1,\ldots,n\} \times \{1,\ldots,n\}}$. These two functions are defined as mutually recursive fixpoints over formulae and programs, respectively. The second part of the semantics amounts to a relation identifying successor states: if $(i, j) \in \mathcal{R}(\alpha, \mathsf{K}^n)$, then there is an α-path from state i to state j in K^n. Looking at the variable case, we *take a v-step* from state i to $i{+}1$ when $\mathsf{K}^n_i(v) = \mathfrak{t}$; other cases are defined with the conventional regular semantics. We define the program $\mathsf{any} = \sum_{v \in \mathsf{Var}} v$ to mean "any possible step". (For any given formula or set of models, we can restrict an infinite Var to a relevant finite subset.) The step relation $\mathcal{R}(\alpha, \mathsf{K}^n)$ does not enforce sensibility of its step indices—that is, it can produce indices past n. We make sure we only use reasonable indices in the definition for $\mathsf{K}^n_i(\langle \alpha \rangle\, \phi)$. We lift the function on formulas to satisfiablity and validity in the usual way (Definition 2).

Let us consider each of the example formulae above in the model K^4 from Section 3. First, ϕ is satisfiable in K^4 because we can take an x-step to find y. Next, ψ is satisfiable in K^4 as well, because there are no z-steps. The formula θ is also satisfiable—we can only take an x-step from the first state, but then y indeed holds. Finally, χ is also satisfiable—perhaps surprisingly so. Why is it the case that $\mathsf{K}^4_1(\neg \langle \mathsf{any}^* \rangle\, z) = \mathfrak{t}$ when $\eta_4(z) = \mathfrak{t}$? We set $\eta_3 = \lambda x.\ \mathfrak{f}$, i.e., no primitive propositions hold in the third time step. It is not possible to take an any-step from η_3 to η_4, so the definition for $\mathcal{R}(\mathsf{any}^*, \mathsf{K}^4)$ will never look past η_3.

LDL_f's semantics are particularly subtle in states where no propositions hold at all: such states break the timeline into multiple, separate fragments. Fragmented timelines effectively truncate the model when thinking about satisfiability (which looks for a formula to hold only from the first state); fragmented timelines encapsulate multiple timelines when thinking about validity (which looks for a formula to hold in every state). In general, it is unsafe to assume that $(i, i+1) \in \mathcal{R}(\mathsf{any}, \mathsf{K}^n)$, even for $i < n$; we are particularly careful while proving that FINITE is sound to recognize that end might come at times other than the nth state.

We show LDL_f's axioms (Figure 4) are sound with respect to the model.

Theorem 17 (Soundness). *If $\vdash \phi$ then $\models \phi$.*

Proof. By induction on the derivation of $\vdash \phi$.

Proof theory $\boxed{\vdash \subseteq 2^{\mathsf{Formula}} \times \mathsf{Formula}}$

Axioms

all propositional tautologies	TAUT
$\vdash \langle a \rangle \neg\phi \Rightarrow \neg \langle a \rangle \phi$	MODALNEGCOMM
$\vdash \langle \alpha + \beta \rangle \phi \Leftrightarrow \langle \alpha \rangle \phi \vee \langle \beta \rangle \phi$	FORMORDISTR
$\vdash \langle \alpha \rangle (\phi \vee \psi) \Leftrightarrow \langle \alpha \rangle \phi \vee \langle \alpha \rangle \psi$	MODALORDISTR
$\vdash \langle \alpha^* \rangle \phi \Leftrightarrow \phi \vee \langle \alpha \rangle \langle \alpha^* \rangle \phi$	STARUNROLL
$\vdash \langle \alpha; \beta \rangle \phi \Leftrightarrow \langle \alpha \rangle \langle \beta \rangle \phi$	MODALSEQ
$\vdash \langle \mathsf{any}^* \rangle \mathsf{end}$	FINITE
$\vdash \langle a \rangle \phi \Rightarrow [b] \phi$	UNIQUESUCC
$\dfrac{\vdash \phi \Rightarrow [\alpha] \phi}{\vdash \phi \Rightarrow [\alpha^*] \phi}$	INDUCTION
$\dfrac{\vdash \phi}{\vdash [v] \phi}$	VARGENERALIZE

Consequences

$\vdash \mathsf{end} \Rightarrow \neg \langle a \rangle \phi$	ENDCONTRA
$\vdash \mathsf{end} \wedge \langle \alpha^* \rangle \phi \Rightarrow \phi$	ENDSTAR
$\dfrac{\vdash \phi}{\vdash [\alpha] \phi}$	GENERALIZE
$\vdash \langle a \rangle \phi \vee \mathsf{end} \Leftrightarrow [a] \phi$	COMMNEG
$\vdash [\alpha^*] \phi \Leftrightarrow \phi \vee [\alpha] [\alpha^*] \phi$	SQUAREUNROLL
$\vdash \neg \langle a \rangle \bot$	MODALCONTRA

Fig. 4. LDL_f proof theory

5.1 Completeness

Our proof of completeness for LDL_f follows the same general structure as that for LTLf (Section 4.1): we define positive-negative pairs, construct a graph, finally using the graph to produce a Kripke structure in a satisfiability lemma. The situation for LDL_f is slightly complicated by its more refined notion of "successor": in LTL_f, every state either has a successor or it is the final state; in LDL_f, a state may have an a successor but not a b successor. Our proofs change accordingly: where we would have used $\bullet\,\phi$ in LTL_f, we use $[a]\,\phi$ for an arbitrary $a \in \mathsf{Var}$; in the last state, we cannot make an any transition, i.e., $\neg \langle \mathsf{any} \rangle \top$.

Following the proof structure of Section 4.1, we construct a proof graph of PNPs using adapted comps and σ functions. We omit the definition of LDL_f PNPs—they correspond exactly to LTL_f's PNPs (Definition 4). In the remainder of the paper, we omit proofs for LDL_f when they correspond more or less directly to those for LTL_f.

Just as in Section 4.1, we define completions (using a variation on the Fischer-Ladner closure, which handles star unrollings better than τ (Figure 2), omitted for space [8]) and successor functions for LDL_f terms (Figure 5). The definitions of assignments, completions, and extensions mirror those for LTL_f.

Lemma 18 (Consistent completions are provable). *For $\mathcal{P}$ a consistent PNP, $\vdash \widehat{\mathcal{P}} \Rightarrow \bigvee_{\mathcal{Q} \in \mathsf{comps}\mathcal{P}} \mathcal{Q}$.*

We define our step function in terms of Brzozowski derivatives [2] with respect to a primitive program $a \in \mathsf{Var}$, which we write $\delta_a(\alpha, \phi)$ (Figure 5). The positive cases for our step function lift a nullability predicate on paths, ν, into our meta-logic, and our result characterizing the derivative of α is predicated on whether or not $\nu(\alpha)$ holds; the negative case need not bother—if $\nu(\alpha)$ and

Step functions and derivatives $\boxed{\sigma_i^\bullet : \mathsf{Var} \times \mathsf{PNP} \to 2^{\mathrm{LDL}_f}}$ $\boxed{\nu \subseteq \mathsf{Program}}$

$\boxed{\delta_a : \mathsf{Program} \times \mathsf{Formula} \to 2^{\mathsf{Formula}}}$

$$\sigma_1^+(a,\mathcal{P}) = \{\psi \mid \langle\alpha\rangle\,\phi \in \mathsf{pos}(\mathcal{P}), \psi \in \delta_a(\alpha,\phi), \text{not } \nu(\alpha)\}$$
$$\sigma_2^+(a,\mathcal{P}) = \{\psi \mid \langle\alpha\rangle\,\phi \in \mathsf{pos}(\mathcal{P}), \psi \in \delta_a(\alpha,\phi), \nu(\alpha), \phi \in \mathsf{neg}(\mathcal{P})\}$$
$$\sigma^-(a,\mathcal{P}) = \{\psi \mid \langle\alpha\rangle\,\phi \in \mathsf{neg}(\mathcal{P}), \psi \in \delta_a(\alpha,\phi)\}$$

$$\frac{}{\nu(a)} \qquad \frac{\nu(\alpha) \qquad \nu(\beta)}{\nu(\alpha;\beta)} \qquad \frac{\nu(\alpha)}{\nu(\alpha+\beta)} \qquad \frac{\nu(\beta)}{\nu(\alpha+\beta)} \qquad \frac{}{\nu(\alpha^*)}$$

$$\begin{aligned}
\delta_a(a,\phi) &= \{\phi\} \\
\delta_a(b,\phi) &= \emptyset \\
\delta_a(\alpha;\beta,\phi) &= \begin{cases} \{\psi \vee \chi \mid \psi \in \delta_a(\alpha,\langle\beta\rangle\,\phi), \chi \in \delta_a(\beta,\phi)\} & \nu(\alpha) \\ \{\psi \mid \psi \in \delta_a(\alpha,\langle\beta\rangle\,\phi)\} & \text{otherwise} \end{cases} \\
\delta_a(\alpha+\beta,\phi) &= \{\psi \vee \chi \mid \psi \in \delta_a(\alpha,\phi), \chi \in \delta_a(\beta,\phi)\} \\
\delta_a(\alpha^*,\phi) &= \delta_a(\alpha,\langle\alpha^*\rangle\,\phi)
\end{aligned}$$

Fig. 5. Closure function; step functions and derivations

$\langle\alpha\rangle\,\phi \in \mathsf{neg}(\mathcal{P})$, it must be the case that $\phi \in \mathsf{neg}(\mathcal{P})$ in any case. As a first step toward understanding our step function, we characterize the derivative.

Lemma 19 (Brzozowski partial equivalence).
For a program α and a formula ϕ,

- *if $\nu(\alpha)$, then $\vdash \langle\alpha\rangle\,\phi \Leftrightarrow \phi \vee \bigwedge_{a\in\mathsf{Var}} \bigwedge_{\psi\in\delta_a(\alpha,\phi)} \langle a\rangle\,\psi$;*
- *if not $\nu(\alpha)$, then $\vdash \langle\alpha\rangle\,\phi \Leftrightarrow \bigwedge_{a\in\mathsf{Var}} \bigwedge_{\psi\in\delta_a(\alpha,\phi)} \langle a\rangle\,\psi$.*

Proof. By induction on the structure of α, leaving ϕ general.

We adapt LTL_f's proof graphs to record labels on each transition.

Definition 20 (Labeled proof graph). *For a consistent and complete PNP $\mathcal{P}$, we define the* labeled proof graph $\mathcal{G}_\mathcal{P}$ *as follows: (a) $\mathcal{P}$ is the root; (b) for each $a \in \mathsf{Var}$, there is an a-edge from $\mathcal{P}$ to $\mathcal{Q}$ for each $\mathcal{Q} \in \mathsf{comps}(\sigma(a,\mathcal{P}))$.*

Having constructed our labeled proof graph, we must find a terminal path—redefining 'terminal' to use LDL_f's notion of end. Once we have the path, we can use it to construct a Kripke structure.

For LTL_f, we show that injecting $\Diamond\,\mathsf{end}$ guarantees that every PNP in the graph has $\Diamond\,\mathsf{end}$ in its positive set. We can prove a similar lemma for LDL_f, being more careful about successors.

Lemma 21 (end injection is weakly invariant). *If $\mathcal{P}$ is a consistent and complete PNP with $\langle\mathsf{any}^*\rangle\,\mathsf{end} \in \mathsf{pos}(\mathcal{P})$, then either:*

– $\neg \langle a \rangle \top \in \mathsf{pos}(\mathcal{P})$ *for every* a *and* $\mathcal{P}$ *has no successors (i.e.* $\mathsf{comps}(\sigma(a, \mathcal{P}) = \emptyset)$ *for all* $a \in \mathsf{Var}$*), or*
– $\langle a \rangle \top \in \mathsf{neg}(\mathcal{P})$ *for some* a *and for all* $\mathcal{Q} \in \mathsf{comps}(\sigma(\mathcal{P}))$, $\langle \mathsf{any}^* \rangle \, \mathsf{end} \in \mathsf{pos}(\mathcal{Q})$.

In showing that successors always exist, we use an arbitrary primitive predicate a—because $[a]\,\phi$ implies $[\mathsf{any}]\,\phi$.

Lemma 22 (Step Implication). *For all consistent and complete PNPs* $\mathcal{P}$ *where* $\langle \mathsf{any}^* \rangle \, \mathsf{end} \in \mathsf{pos}(\mathcal{P})$*,, if* $\mathcal{Q} \in \mathcal{G}_{\mathcal{P}}$ *then for every primitive program* a*,* $\vdash \widehat{\mathcal{Q}} \Rightarrow [a]\, \widehat{\sigma(a, \mathcal{Q}_i)}$.

Lemma 23 (All non-terminal PNPs have successors in the labeled graph). *For all consistent and complete PNPs* $\mathcal{P}$*, if* $\langle \mathsf{any}^* \rangle \, \mathsf{end} \in \mathsf{pos}(\mathcal{P})$*, then for every* $a \in \mathsf{Var}$*, we have* $\vdash \left(\bigvee_{\mathcal{Q} \in V(\mathcal{G}_{\mathcal{P}})} \widehat{\mathcal{Q}} \right) \Rightarrow [a] \left(\bigvee_{\mathcal{Q} \in V(\mathcal{G}_{\mathcal{P}})} \widehat{\mathcal{Q}} \right)$.

Corollary 24 (Terminal path existence). *For all consistent and complete PNPs* $\mathcal{P}$ *where* $\langle \mathsf{any}^* \rangle \, \mathsf{end} \in \mathsf{pos}(\mathcal{P})$*, the labeled proof graph* $\mathcal{G}_{\mathcal{P}}$ *has a terminal path.*

We omit (due to space constraints) proofs showing that the labeled proof graph adequately generates Kripke structures; our proofs show that (a) the labeled proof graph has an appropriate relationship to programs, and (b) that the graph respects temporal operators. From here, the proof is the usual one: prove satisfiability and then completeness.

Theorem 25 (LDL$_f$ satisfiability). *If* $\mathcal{P}$ *is a consistent PNP, then* $\widehat{\mathcal{P}}$ *is satisfiable.*

Theorem 26 (LDL$_f$ completeness). *If* $\models \phi$ *then* $\vdash \phi$.

6 Discussion

We have defined two finite temporal logics for linear time: LTL$_f$ and LDL$_f$. The two logics share models, though their proof theories differ. In both cases, we were able to adapt techniques for *infinite* temporal logics to show deductive completeness in a finite setting. The former proof improves on Roşu's axioms [12]; the latter proof is novel. In both cases, the proofs of deductive completeness call for only minor changes to the proof for logics with potentially infinite time: by *injecting finiteness*, we are able to directly adapt methods from infinite logics. We believe that the technique is general, and will adapt to other temporal logics; we offer these two proofs as evidence.

To be clear, we claim that the proof of completeness is relatively straightforward *once you find the right axioms*. We can offer only limited guidance on finding the right axioms. Finite temporal logics should have an axiom saying that time is, indeed, finite; some sort of axiom will be needed to establish the meaning of temporal modalities at the end of time; when porting axioms from the infinite logic, one must be careful to check that the axioms are sound at the end of time, when temporal modalities may change in meaning.

References

1. Baier, J.A., McIlraith, S.A.: Planning with first-order temporally extended goals using heuristic search. In: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1. pp. 788–795. AAAI'06, AAAI Press (2006), http://dl.acm.org/citation.cfm?id=1597538.1597664
2. Brzozowski, J.A.: Derivatives of regular expressions. J. ACM 11(4), 481–494 (Oct 1964), http://doi.acm.org/10.1145/321239.321249
3. D'Antoni, L., Veanes, M.: Monadic second-order logic on finite sequences. In: Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages. pp. 232–245. POPL 2017, ACM, New York, NY, USA (2017), http://doi.acm.org/10.1145/3009837.3009844
4. De Giacomo, G., De Masellis, R., Montali, M.: Reasoning on ltl on finite traces: Insensitivity to infiniteness. In: AAAI. pp. 1027–1033. Citeseer (2014)
5. De Giacomo, G., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. pp. 854–860. Association for Computing Machinery (2013)
6. De Giacomo, G., Vardi, M.Y.: Synthesis for ltl and ldl on finite traces. In: Proc. of IJCAI (2015)
7. De Giacomo, G., Vardi, M.Y.: Ltlf and ldlf synthesis under partial observability. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. pp. 1044–1050. IJCAI'16, AAAI Press (2016), http://dl.acm.org/citation.cfm?id=3060621.3060766
8. Fischer, M.J., Ladner, R.E.: Propositional dynamic logic of regular programs. Journal of computer and system sciences 18(2), 194–211 (1979)
9. Kroger, F., Merz, S.: Temporal logic and state systems. Springer (2008)
10. Lichtenstein, O., Pnueli, A., Zuck, L.: The glory of the past. In: Workshop on Logic of Programs. pp. 196–218. Springer (1985)
11. Pnueli, A.: The temporal logic of programs. In: Foundations of Computer Science, 1977., 18th Annual Symposium on. pp. 46–57 (Oct 1977)
12. Roşu, G.: Finite-trace linear temporal logic: Coinductive completeness. In: International Conference on Runtime Verification. pp. 333–350. Springer (2016)
13. Vardi, M.Y.: The rise and fall of linear temporal logic. In: Proceedings of the 2nd International Syposium on Games, Automata, Logics, and Formal Verification (11)