dTL^2: Differential Temporal Dynamic Logic with Nested Temporalities for Hybrid Systems

Jean-Baptiste Jeannin **André Platzer**

December 2014
CMU-CS-14-109

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

The differential temporal dynamic logic dTL^2 is a logic to specify temporal properties of hybrid systems. It combines differential dynamic logic with temporal logic to reason about the intermediate states reached by a hybrid system. The logic dTL^2 supports some linear time temporal properties of LTL. It extends differential temporal dynamic logic dTL with nested temporalities. We provide a semantics and a proof system for the logic dTL^2, and show its usefulness for nontrivial temporal properties of hybrid systems. We take particular care to handle the case of alternating universal dynamic and existential temporal modalities and its dual, solving an open problem formulated in previous work.

This material is based upon work supported by the National Science Foundation under NSF CAREER Award CNS-1054246, NSF EXPEDITION CNS-0926181 and under Grant No. CNS-0931985. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution or government. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of any sponsoring institution or government.

Keywords: differential temporal dynamic logic; hybrid systems; dynamic logic; temporal logic

1 Introduction

A major task of computer science is to program objects of our physical world: cars, trains, airplanes, robots, etc. — often grouped under the denomination of cyber-physical systems (CPS). A CPS is governed by its programmable controllers, but also by the laws of physics. To fully verify it, one thus needs to model the controllers and their software as well as the relevant laws of physics in the same system. Such a system then becomes hybrid: the controllers are *discrete* while the laws of physics are *continuous*.

In recent years, a number of systems have been explored to reason about such *hybrid systems*. In particular, this paper is based on *differential dynamic logic* [Pla07], [Pla10, chapter 4], a logic based on dynamic logic [Pla08, Pla12], [Pla10, chapter 2] and including programs enabling discrete assignments and discrete control structures, but also execution of differential equations. Differential dynamic logic comes with a semantics as well as a proof system, which is sound and relatively complete.

Based on dynamic logic, differential dynamic logic only reasons about the end state of a system. However, to ensure that a system always stays within some structural limits, or always accomplishes a certain task, one needs to reason about its intermediate states as well. CPSs that are safe when their systems terminate but have been unsafe in the middle of the program run are still not safe to use. The idea is to use both dynamic logic — to quantify over possible executions — and temporal logic — to quantify over the states in the trace of each execution. This is not a new idea, but previous work [BS01, Pla07] focuses only on the non-alternating cases: "some property is always verified during all executions" and "something happens during some execution."

In this paper, we are developing a *differential temporal dynamic logic* dTL^2 inspired from LTL, and we are focusing on correctly handling the more complex alternating cases: "something happens during all executions" and "there is an execution where some property is always verified," as well as nested temporal modalities. In particular, a property checking that a task is always accomplished can now be checked. This logic is an important stepping stone towards full dTL*, the differential analog of CTL*.

As a simple example, let us look at a satellite with position x trying to leave the solar system, avoiding planets. To simplify, let us consider only two planets with radiuses r_1 and r_2, at (evolving) positions p_1 and p_2. The satellite can be controlled either by a pilot who can set its steering ω to left or right then let x evolve according to differential equation $\mathsf{flight}(\omega)$, or by an autopilot following a PID controller with target direction set to d. During each evolution, the positions of the planets continue to evolve, following differential equation $\mathsf{planets}(p_1, p_2)$. The program of the satellite and its safety property ϕ — expressing that there exists a steering avoiding all planets — can be expressed as:

$$\begin{aligned}\mathsf{satellite} ::= &(((\omega := \mathsf{left} \cup \omega := \mathsf{right}); x' = \mathsf{flight}(\omega), (p_1', p_2') = \mathsf{planets}(p_1, p_2)) \\ &\cup (d := *; x' = \mathsf{PID}(d), (p_1', p_2') = \mathsf{planets}(p_1, p_2)))^*; \\ &\mathsf{control} := \mathsf{lost}; d := *; x' = \mathsf{PID}(d), (p_1', p_2') = \mathsf{planets}(p_1, p_2) \\ \phi ::= &\langle \mathsf{satellite} \rangle \Box (\mathsf{dist}(x, p_1) > r_1 \wedge \mathsf{dist}(x, p_2) > r_2 \wedge \mathsf{control} \neq \mathsf{lost})\end{aligned}$$

This example shows several features of hybrid programs and the logic dTL^2. Under the pilot's com-

mand, the variable ω can be *assigned* to either left or right, following a *nondeterministic choice* $\cup$. Then x, p_1 and p_2 follow a *differential equation* modeling the continuous evolution of the system, including movement of the planets. Under the autopilot's command, d is *nondeterministically assigned* $(d := *)$. There is a nondeterministic choice between the two commands, followed by a star * representing *repetition*. In case of mechanical or communication failure, control could be lost, which we represent by a variable assignment, and the system continues to evolve. The formula ϕ says that there exists a possible evolution ($\langle \mathsf{satellite} \rangle$) such that throughout this evolution ($\Box$), the satellite does not hit any planet; namely, the evolution avoiding planets where control is never lost. The formula ϕ is expressible in dTL^2, and shows how dTL^2 handles alternating and nested program ($\langle \mathsf{satellite} \rangle$) and temporal modalities ($\Box$ and $\Diamond$). The focus of this paper is to create a semantics and a proof calculus for dTL^2.

There are three main contributions to this paper. First, we show how to correctly handle the alternating cases of a universal dynamic modality followed by an existential temporal modality, and its dual an existential dynamic modality followed by a universal temporal modality. This solves an open problem identified in 2001 [BS01] and identified as a problem for hybrid systems in 2007 [Pla07], [Pla10, chapter 4]. Second, we offer a treatment where programs are not duplicated by proof rules, solving another open problem formulated in [Pla07], [Pla10, chapter 4]. This is significant for proving hybrid systems in practice, because previous approaches led to a duplication of proof effort, once for intermediate and once for final states. Third and finally, we extend the logic to nested temporal quantifiers, show that all formulas of interest are equivalent to formulas containing at most two quantifiers — thus the name dTL^2 — by identifying the resemblance to modal system S4.2, and develop a logic and proof calculus for the new temporal formulas.

The paper is organized as follows. After presenting the syntax and semantics of Differential Temporal Dynamic Logic dTL^2 in Section 2, we show how to normalize trace formulas and how to axiomatize dTL^2 in Section 3. We study alternative proof systems in Section 4 and related work in Section 5, before concluding in Section 6.

2 Differential Temporal Dynamic Logic dTL^2

This section defines the syntax and semantics of hybrid programs and trace formulas formally. The development mostly follows and extends previous work on differential temporal dynamic logic [Pla07], [Pla10, chapter 4]; we explicitly point out differences and extensions from the previous work.

2.1 Hybrid Programs

We use *hybrid programs* (HP) [Pla08, Pla12], [Pla10, chapter 2] α, β to model hybrid systems. Syntactically, hybrid programs can be *atomic* hybrid programs or *compound* hybrid programs. Atomic hybrid programs can be discrete jump *assignments* ($x := \theta$), *tests* ($?\chi$) and *differential equations* evolving within an evolution domain constraint χ — meaning that the system can evolve following a solution of the differential equation as long as χ remains true ($x' = \theta \,\&\, \chi$). Terms θ are

polynomials with rational coefficients, and conditions χ are first-order formulas of real arithmetic.[1] Compound hybrid programs are *nondeterministic choice* $(\alpha \cup \beta)$, *sequential composition* $(\alpha; \beta)$ and *nondeterministic finite repetition* (α^*):

$$\alpha, \beta ::= x := \theta \mid ?\chi \mid x' = \theta \,\&\, \chi \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$$

The *trace semantics* of hybrid programs assigns to each program α a set of *traces* $\tau(\alpha)$. The set of *states* Sta is the set of (total) functions from variables to the reals $\mathbb{R}$. In addition, we consider a separate state Λ (not in Sta) denoting a failure of the system. For $v \in \mathsf{Sta}$ or $v = \Lambda$, we denote by $\hat{v}$ the function $\sigma : \{0\} \to \{v\}, 0 \mapsto v$, defined only on the singleton interval $[0, 0]$. A *trace* is a (nonempty) finite sequence $\sigma = (\sigma_0, \sigma_1, ..., \sigma_n)$ of functions σ_i. For $0 \leq i < n$, the piece σ_i is a function $\sigma_i : [0, r_i] \to \mathsf{Sta}$, where $r_i \geq 0$ is the duration of this step. For $i = n$, the piece σ_n is either a function:

- $\sigma_n : [0, r_i] \to \mathsf{Sta}$; we then say that σ is a *terminating* trace; or
- $\sigma_n : [0, +\infty) \to \mathsf{Sta}$; we then say that σ is an *infinite* trace; or
- $\sigma_n : \{0\} \to \{\Lambda\}, 0 \mapsto \Lambda$, for $n \geq 1$;[2] we then say that σ is an *error* trace.

We often collectively refer to infinite and error traces as *nonterminating*; thus when we refer to terminating traces, we only refer to those traces that terminate but not with an error state Λ. We write Tra for the set of all traces. A *position* of σ is a pair (i, ζ) with $0 \leq i \leq n$ and ζ in the domain of definition of σ_i; the state of σ at (i, ζ) is $\sigma_i(\zeta)$. For any trace σ, we denote by $\mathsf{first}\,\sigma$ the state $\sigma_0(0)$; we informally say that "σ starts with v" to say that $v = \mathsf{first}\,\sigma$. If $\sigma = (\sigma_0, \ldots, \sigma_n)$ terminates (and only in that case), we also denote by $\mathsf{last}\,\sigma$ the state $\sigma_n(r_n)$; when σ does not terminate, $\mathsf{last}\,\sigma$ is undefined. We denote by $\mathsf{val}(v, \theta)$ the value of term θ in state v, and by $v[x \mapsto r]$ the valuation assigning variable x to $r \in \mathbb{R}$ and matching with v on all other variables. We also write $v \vDash \chi$ if state v satisfies condition χ, and $v \nvDash \chi$ otherwise.

Given two traces $\sigma = (\sigma_0, \ldots, \sigma_n)$ and $\rho = (\rho_0, \ldots, \rho_m)$, we say that ρ is a *prefix* of σ if it describes the trace σ truncated at some position. Formally, ρ is a prefix of σ if and only if $\rho = \sigma$ — a condition ensuring that nonterminating traces are also suffixes of themselves — or there exists a position (i, ζ) of σ such that:

- traces $(\sigma_0, \ldots, \sigma_{i-1})$ and $(\rho_0, \ldots, \rho_{m-1})$ are identical.[3] In particular this imposes that $i = m$; and
- the domain of definition of ρ_m is exactly $[0, \zeta]$ and is included in the domain of definition of σ_m, and for all $d \in [0, \zeta]$, $\sigma_m(d) = \rho_m(d)$.

[1] using first-order formulas or real arithmetic results in a poor-test version of the logic. Our results generalize to a rich-test version, where a condition χ is instead defined as any formula ϕ of dTL^2 (see Section 2.2).

[2] We impose $n \geq 1$ so that $(\hat{\Lambda})$ is not considered a trace

[3] if $i = m = 0$, $(\sigma_0, \ldots, \sigma_{i-1})$ and $(\rho_0, \ldots, \rho_{m-1})$ are empty and thus not formally traces, but we still consider the condition fulfilled.

Symmetrically, we say that ρ is a *suffix* of σ if it starts at some position of σ then follows σ. Formally, ρ is a suffix of σ if and only if there exists a position (i, ζ) of σ such that:

- if σ_i has domain of definition $[0, r_i]$, then the domain of definition of ρ_0 is exactly $[0, r_i - \zeta]$ and for all $d \in [\zeta, r_i]$, $\sigma_i(d) = \rho_0(d - \zeta)$; and in the case where σ_i has domain of definition $[0, +\infty)$, the domain of definition of ρ_0 is also $[0, +\infty)$ and for all $d \in [\zeta, +\infty)$, $\sigma_i(d) = \rho_0(d - \zeta)$; and
- $(\sigma_{i+1}, \ldots, \sigma_n)$ and $(\rho_1, \ldots, \rho_m)$ are identical, which imposes that $n - i = m$.

Definition 1 (Trace Semantics of Hybrid Programs)**.** The *trace semantics* $\tau(\alpha) \subseteq 2^{\mathsf{Tra}}$ of a hybrid program α is then defined inductively as follows:

- $\tau(x := \theta) = \{(\hat{v}, \hat{w}) \mid w = v[x \mapsto \mathsf{val}(v, \theta)]\}$;
- $\tau(x' = \theta \,\&\, \chi) = \{(\sigma) \ : \ \sigma$ is a state flow of order 1 [Pla08] defined on $[0, r]$ or $[0, +\infty)$ solution of $x' = \theta$, and for all t in its domain of definition, $\sigma(t) \vDash \chi\}$
  $\cup \{(\hat{v}, \hat{\Lambda}) \ : \ v \nvDash \chi\}$;[4]
- $\tau(?\chi) = \{(\hat{v}) \ : \ v \vDash \chi\} \cup \{(\hat{v}, \hat{\Lambda}) \ : \ v \nvDash \chi\}$;
- $\tau(\alpha \cup \beta) = \tau(\alpha) \cup \tau(\beta)$;
- $\tau(\alpha; \beta) = \{\sigma \circ \rho \ : \ \sigma \in \tau(\alpha), \rho \in \tau(\beta)$ when $\sigma \circ \rho$ is defined$\}$;
  where the composition $\sigma \circ \rho$ of $\sigma = (\sigma_0, \ldots, \sigma_n)$ and $\rho = (\rho_0, \ldots, \rho_m)$ is
  - $\sigma \circ \rho = (\sigma_0, \ldots, \sigma_n, \rho_0, \ldots, \rho_m)$ if σ terminates and $\mathsf{last}\ \sigma = \mathsf{first}\ \rho$ (since σ terminates, $\mathsf{last}\ \sigma$ is well-defined);
  - σ if σ does not terminate;
  - undefined otherwise;
- $\tau(\alpha^*) = \bigcup_{n \in \mathbb{N}} \tau(\alpha^n)$, where α^0 is defined as $?\mathsf{true}$, α^1 is defined as α and α^{n+1} is defined as $\alpha^n; \alpha$ for $n \geq 1$.

An important property of this trace semantics is that for all programs α and states v, there exists a trace σ of α starting with v (even if it might be an error trace). This property will be key to proving the soundness of assignment rules.

Aside from the correction on $\tau(x' = \theta \,\&\, \chi)$, this definition is slightly different from [Pla07], [Pla10, chapter 4] in two ways: these previous papers also consider infinite sequences $\sigma = (\sigma_0, \sigma_1, \ldots)$, but infinite sequences are not part of the semantics of any program; and these papers do not consider infinite traces in the semantics. Still, we can prove that the interpretation of trace formulas (Section 2.2) is the same on the subset of trace formulas they consider.

[4] this case is corrected from [Pla07], [Pla10, chapter 4], which wrongly forget the error traces of ordinary differential equations — when χ is initially false.

2.2 State and Trace Formulas

To reason about hybrid programs, we use *state formulas* and *trace formulas*. State formulas express properties about states, while trace formulas express properties about traces; their definitions are mutually inductive. A state formula ϕ, ψ can be a *comparison of terms* ($\theta_1 \geq \theta_2$); a *negation* of a state formula ($\neg\phi$); a *conjunction* ($\phi \wedge \psi$) or a *disjunction* ($\phi \vee \psi$) of state formulas; a *universally quantified* ($\forall x\ \phi$) or *existentially quantified* ($\exists x\ \phi$) state formula — quantification of a variable x is over the set of reals $\mathbb{R}$. Finally, a state formula can also be a *program necessity* ($[\alpha]\pi$) — expressing that all traces of hybrid program α starting at the current state satisfy trace formula π — or its dual, a *program possibility* ($\langle\alpha\rangle\pi$) — expressing that there is a trace of α starting at the current state satisfying trace formula π.

A trace formula π can be a *state formula* (ϕ); a *negation* of a trace formula ($\neg\pi$); a *temporal necessity* of a trace formula ($\Box\pi$) — expressing that every suffix of the current trace satisfies π — or its dual, a *temporal possibility* of a trace formula ($\Diamond\pi$) — expressing that there is a suffix of the current trace satisfying π. The syntax of state and trace formulas is thus given by:

$$\begin{aligned} \phi, \psi &::= \theta_1 \geq \theta_2 \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \forall x\ \phi \mid \exists x\ \phi \mid [\alpha]\pi \mid \langle\alpha\rangle\pi \\ \pi &::= \phi \mid \neg\pi \mid \Box\pi \mid \Diamond\pi \end{aligned}$$

Additionally, as in classical logic, the implication $\phi \rightarrow \psi$ is defined as $\neg\phi \vee \psi$. When a trace formula also happens to be a state formula ϕ, the formula $\neg\phi$ means the same whether it is seen as a state or trace formula; in the rest of the paper we collude the two. We are now ready to define satisfaction of state and trace formulas.

Definition 2 (Satisfaction of dTL^2 Formulas)**.** For state formulas, we write $v \vDash \phi$ to say that state $v \in \mathsf{Sta}$ satisfies state formula ϕ. Satisfaction of state formulas with respect to a state v is defined inductively as follows:

- $v \vDash \theta_1 \geq \theta_2$ if and only if $\mathsf{val}(v, \theta_1) \geq \mathsf{val}(v, \theta_2)$
- $v \vDash \neg\phi$ if and only if $v \vDash \phi$ does not hold.
- $v \vDash \phi \wedge \psi$ if and only if $v \vDash \phi$ and $v \vDash \psi$.
- $v \vDash \phi \vee \psi$ if and only if $v \vDash \phi$ or $v \vDash \psi$.
- $v \vDash \forall x\, \phi$ if and only if $v[x \mapsto d] \vDash \phi$ holds for all $d \in \mathbb{R}$.
- $v \vDash \exists x\, \phi$ if and only if $v[x \mapsto d] \vDash \phi$ holds for some $d \in \mathbb{R}$.
- for ϕ a state formula, $v \vDash [\alpha]\phi$ if and only if for each trace $\sigma \in \tau(\alpha)$ that starts in $\mathsf{first}\ \sigma = v$, if σ terminates, then $\mathsf{last}\ \sigma \vDash \phi$.
- for ϕ a state formula, $v \vDash \langle\alpha\rangle\phi$ if and only if there is a trace $\sigma \in \tau(\alpha)$ starting in $\mathsf{first}\ \sigma = v$ such that σ terminates and $\mathsf{last}\ \sigma \vDash \phi$.

- If π is not a state formula, $v \vDash [\alpha]\pi$ if and only $\sigma \vDash \pi$ for each trace $\sigma \in \tau(\alpha)$ that starts in $\mathsf{first}\ \sigma = v$.
- If π is not a state formula, $v \vDash \langle\alpha\rangle\pi$ if and only $\sigma \vDash \pi$ for some trace $\sigma \in \tau(\alpha)$ that starts in $\mathsf{first}\ \sigma = v$.

For trace formulas, we write $\sigma \vDash \pi$ to say that trace $\sigma \in \mathsf{Tra}$ satisfies trace formula π. Satisfaction of trace formulas with respect to a trace σ is defined inductively as follows:

- $\sigma \vDash \phi$ if and only if $\mathsf{first}\ \sigma \vDash \phi$.
- $\sigma \vDash \neg\pi$ if and only if $\sigma \vDash \pi$ does not hold.
- $\sigma \vDash \Box\pi$ if and only if $\rho \vDash \pi$ holds for all suffixes ρ of σ that are different from $(\hat{\Lambda})$.
- $\sigma \vDash \Diamond\pi$ if and only if $\rho \vDash \pi$ holds for some suffix ρ of σ that is different from $(\hat{\Lambda})$.

This definition follows the intuition given when presenting the syntax of state and trace formulas, except for one point. Note that in the definitions of $\sigma \vDash \Box\pi$ and $\sigma \vDash \Diamond\pi$, the suffix ρ of σ does not have to be proper, and we can have $\rho = \sigma$. When seen as a trace formula, a state formula ϕ can express a property on a trace σ. We then say that σ satisfies ϕ if and only if the first state of σ satisfies ϕ (condition $\mathsf{first}\ \sigma \vDash \phi$ in the definition of $\sigma \vDash \phi$). However, there is an exception to this definition: when ϕ appears directly after a program necessity (as in $[\alpha]\phi$) or a program possibility (as in $\langle\alpha\rangle\phi$), ϕ only refers to *terminating* traces, and we say that σ satisfies ϕ if and only if the *last* state of σ satisfies ϕ (condition $\sigma \vDash \mathsf{last}\ \phi$ in the definitions of $\sigma \vDash \langle\alpha\rangle\phi$ and $\sigma \vDash [\alpha]\phi$). This discontinuity in the definition of the satisfaction of ϕ enables following both the usual semantics of dynamic logic and of temporal logic, and was also adopted in previous work [HKP82, Pla07], [Pla10, chapter 4]. It is also useful for proof rules as temporal properties often reduce to what happens after a program.

The syntax of dTL^2 formulas extends the syntax of trace formulas given in [Pla07], [Pla10, chapter 4] by allowing nesting of temporal modalities, and otherwise agrees with it. The satisfaction of dTL^2 formulas given in Def. 2, although presented in a slightly different way, agrees with the definitions given in [Pla07], [Pla10, chapter 4] on trace formulas without nested temporal modalities.

3 Proof Calculus

3.1 Equivalence of Trace Formulas

Trace formulas follow the axioms of modal system S4.2 [HC96], therefore there are only four proper affirmative modalities $\Box\phi$, $\Diamond\phi$, $\Box\Diamond\phi$ or $\Diamond\Box\phi$. Intuitively, because formulas $\neg\Box\pi$ and $\Diamond\neg\pi$ are equivalent — in the sense that they are satisfied by the same traces — formulas can always be expressed in a way where only state formulas have negations. Similarly, formulas $\Box\pi$ and $\Box\Box\pi$ are equivalent, therefore a trace formula containing exclusively $\Box$ temporalities followed by a state formula ϕ is equivalent to $\Box\phi$. Moreover, a formula containing both $\Box$ and $\Diamond$ temporalities,

finishing by a $\Diamond$ temporality followed by a state formula ϕ is equivalent to $\Box\Diamond\phi$. Similar properties are true for their duals. This is formalized by the following lemma, proved in Appendix B.

Lemma 1 (Equivalence of Trace Formulas)**.** *For any trace formula π_1, there exists a trace formula π_2 of the form ϕ, $\Box\phi$, $\Diamond\phi$, $\Box\Diamond\phi$ or $\Diamond\Box\phi$ such that $\sigma \vDash \pi_1$ if and only if $\sigma \vDash \pi_2$. Such a π_2 can be computed from π_1 in linear time in the number of temporal modalities and negations in π_1.*

Remark 1. Lemma 1 tells us that the only interesting trace formulas of our system are those of the form ϕ, $\Box\phi$, $\Diamond\phi$, $\Box\Diamond\phi$ and $\Diamond\Box\phi$. For any trace σ, the intuitive meaning of $\sigma \vDash \pi$ for π of the form ϕ, $\Box\phi$ or $\Diamond\phi$ is clear: we have $\sigma \vDash \phi$ if and only if σ starts in a state satisfying ϕ; we have $\sigma \vDash \Box\phi$ if and only if all non-error states of the trace σ satisfy ϕ; and we have $\sigma \vDash \Diamond\phi$ if and only if there is a non-error state of trace σ satisfying ϕ. When π is of the form $\Box\Diamond\phi$ and $\Diamond\Box\phi$, we get a better intuition by distinguishing cases:

- if σ is a terminating trace, $\sigma \vDash \Diamond\Box\phi$ if and only if $\mathsf{last}\,\sigma \vDash \phi$, and $\sigma \vDash \Box\Diamond\phi$ if and only if $\mathsf{last}\,\sigma \vDash \phi$ as well;
- if σ is an error trace, σ can be written $(\sigma_0, \ldots, \sigma_{n-1}, \hat{\Lambda})$. Let $\rho = (\sigma_0, \ldots, \sigma_{n-1})$, then ρ is a terminating trace and a prefix of σ. Moreover, both $\sigma \vDash \Diamond\Box\phi$ and $\sigma \vDash \Box\Diamond\phi$ are equivalent to $\mathsf{last}\,\rho \vDash \phi$;
- if σ is an infinite trace, $\sigma \vDash \Diamond\Box\phi$ holds if and only if ϕ holds on all states of σ after some position, and $\sigma \vDash \Box\Diamond\phi$ holds if and only if any state of σ has a later state satisfying ϕ (if we did not have continuous dynamics, this would be the same as ϕ being true infinitely often along σ; but here it is not sufficient).

3.2 Normalization of Trace Formulas

The primary goal of this paper is to establish a proof system for differential temporal dynamic logic dTL^2. As for $\mathsf{d}\mathcal{L}$ and dTL, rules typically decompose programs syntactically. Let us look at the state formula $\langle\alpha;\beta\rangle\Box\phi$, and to simplify, let us only consider terminating traces for now. Intuitively, this formula says that there exists a trace in $\tau(\alpha;\beta)$ throughout which ϕ holds. Considering only terminating traces, this is true as long as there exists a trace σ of α throughout which ϕ is true, and a trace ρ of β starting at $\mathsf{last}\,\sigma$ throughout which ϕ is also true. It is thus tempting to write the following rule:

$$\frac{\langle\alpha\rangle\Box\phi \wedge \langle\alpha\rangle\langle\beta\rangle\Box\phi}{\langle\alpha;\beta\rangle\Box\phi}\ \text{(unsound)}$$

This rule is unsound because α is possibly nondeterministic. Its premise says that there is a trace σ of α throughout which ϕ is true, and a trace σ' of α followed by a trace ρ of β throughout which ϕ is true. But σ and σ' do not have to be the same trace; the trick is that ϕ is not necessarily true throughout σ'. To fix this rule, we need to express that traces σ and σ' are the same, thus writing a premise resembling:

$$\langle\alpha\rangle(\Box\phi \wedge \langle\beta\rangle\Box\phi) \tag{1}$$

Unfortunately, this is not directly expressible with dTL^2, without using the program $\alpha;\beta$ again: the missing piece is the expressibility of a conjunction on traces that simultaneously talks about temporal properties like $\Box\phi$ and properties true at the end of the trace. To achieve this expressibility, we extend the logic with *normalized trace formulas* to make conjunction of temporal formulas expressible as needed in (1).

A normalized trace formula ξ can be of different forms: for terminating traces, the formula $\phi \sqcap \Box\psi$ captures the conjunction of ending in a state satisfying ϕ, and satisfying $\Box\psi$; and the formula $\phi \sqcup \Diamond\psi$ captures the disjunction of ending in a state satisfying ϕ, or satisfying $\Diamond\psi$. For nonterminating traces, $\phi \sqcap \Box\psi$ is the same as $\Box\psi$, and $\phi \sqcup \Diamond\psi$ is the same as $\Diamond\psi$, because there is no terminal state in which it makes sense to evaluate ϕ. Additionally, the formula $\phi \blacktriangleleft \Box\Diamond\psi$ captures ending in a state satisfying ϕ if terminating, and satisfying $\Box\Diamond\psi$ otherwise; and similarly, the formula $\phi \blacktriangleleft \Diamond\Box\psi$ captures ending in a state satisfying ϕ if terminating, and satisfying $\Diamond\Box\psi$ otherwise.

Formulas $\phi \blacktriangleleft \Diamond\Box\psi$ and $\phi \blacktriangleleft \Box\Diamond\psi$ play the same role for formulas $\Diamond\Box\psi$ and $\Box\Diamond\psi$ as formulas $\phi \sqcap \Box\psi$ and $\phi \sqcup \Diamond\psi$ play for formulas $\Box\psi$ and $\Diamond\psi$: they allow us to define premises of modular inference rules for sequential composition as in (1). Like standard trace formulas, normalized trace formulas can appear after a program necessity $[\alpha]$ or a program possibility $\langle\alpha\rangle$. We therefore extend state formulas to accept normalized trace formulas, and define normalized trace formulas as:

$$\begin{aligned} \phi, \psi &::= \ldots \mid [\alpha]\xi \mid \langle\alpha\rangle\xi \\ \xi &::= \phi \sqcap \Box\psi \mid \phi \sqcup \Diamond\psi \mid \phi \blacktriangleleft \Box\Diamond\psi \mid \phi \blacktriangleleft \Diamond\Box\psi \end{aligned}$$

Sometimes we will also use the notation $\phi \blacktriangleleft \pi$, with the understanding that in such cases π can only be of the form $\Box\Diamond\psi$ or $\Diamond\Box\psi$.

Coming back to our example, a sound rule for $\langle\alpha;\beta\rangle\Box\phi$ can be expressed as:

$$\frac{\langle\alpha\rangle(\langle\beta\rangle\Box\phi \sqcap \Box\phi)}{\langle\alpha;\beta\rangle\Box\phi}\ (\langle;\rangle\Box)$$

In the form of its dual $[;]\Diamond$, this rule will be discussed later and proved sound in Appendix A.1. Observe how $\langle;\rangle\Box$ does not even duplicate α and β.

Extending Def. 2, the satisfaction of trace formulas $[\alpha]\xi$ and $\langle\alpha\rangle\xi$ is defined in the same way as trace formulas $[\alpha]\pi$ and $\langle\alpha\rangle\pi$ (if π is not a state formula):

- $v \vDash [\alpha]\xi$ if and only $\sigma \vDash \xi$ for each trace $\sigma \in \tau(\alpha)$ that starts in $\mathsf{first}\ \sigma = v$.
- $v \vDash \langle\alpha\rangle\xi$ if and only $\sigma \vDash \xi$ for some trace $\sigma \in \tau(\alpha)$ that starts in $\mathsf{first}\ \sigma = v$.

Satisfaction of normalized trace formulas carefully distinguishes between terminating and nonterminating traces, and is defined as follows.

Definition 3 (Semantics of Normalized dTL^2 Trace Formulas)**.** For normalized trace formulas, we write $\sigma \vDash \xi$ to say that trace σ satisfies normalized state formula ξ. Satisfaction of normalized trace formulas with respect to a trace σ is defined inductively:

$$\Box\phi \rightsquigarrow \mathsf{true} \sqcap \Box\phi \ (\rightsquigarrow\sqcap) \qquad \Diamond\phi \rightsquigarrow \mathsf{false} \sqcup \Diamond\phi \ (\rightsquigarrow\sqcup)$$

$$\Box\Diamond\phi \rightsquigarrow \phi \blacktriangleleft \Box\Diamond\phi \ (\rightsquigarrow\blacktriangleleft\Box) \qquad \Diamond\Box\phi \rightsquigarrow \phi \blacktriangleleft \Diamond\Box\phi \ (\rightsquigarrow\blacktriangleleft\Diamond)$$

$$\phi \rightsquigarrow \phi \ (\rightsquigarrow\phi) \qquad \frac{\pi_1 \sim \pi_2 \qquad \pi_2 \rightsquigarrow \xi}{\pi_1 \rightsquigarrow \xi} \ (\sim\rightsquigarrow)$$

Figure 1: Normalization rules for trace formulas

$$\sigma \vDash \phi \sqcup \Diamond\psi \quad \text{if and only if} \quad \begin{cases} \mathsf{last}\ \sigma \vDash \phi \text{ or } \sigma \vDash \Diamond\psi & \text{if } \sigma \text{ terminates} \\ \sigma \vDash \Diamond\psi & \text{otherwise} \end{cases}$$

$$\sigma \vDash \phi \sqcap \Box\psi \quad \text{if and only if} \quad \begin{cases} \mathsf{last}\ \sigma \vDash \phi \text{ and } \sigma \vDash \Box\psi & \text{if } \sigma \text{ terminates} \\ \sigma \vDash \Box\psi & \text{otherwise} \end{cases}$$

$$\sigma \vDash \phi \blacktriangleleft \pi \quad \text{if and only if} \quad \begin{cases} \mathsf{last}\ \sigma \vDash \phi & \text{if } \sigma \text{ terminates} \\ \sigma \vDash \pi & \text{otherwise} \end{cases}$$

Not only can normalized trace formulas help express rules like $\langle;\rangle\Box$, they can also, along with state formulas, express all possible trace formulas. In Lemma 1, we have shown how to express any trace formula in the form ϕ, $\Box\phi$, $\Diamond\phi$, $\Box\Diamond\phi$ or $\Diamond\Box\phi$. Building on this result, we now show how to *normalize* every trace formula into a state formula or a normalized trace formula. To this effect, we define a relation $\rightsquigarrow$ between the set of state formulas and trace formulas, and the set of state formulas and normalized trace formulas. This simplifies the axiomatization of dTL^2 by allowing us to only consider cases containing normalized trace formulas.

The normalization is sound, meaning that two related formulas are satisfied by the same trace. Additionally, every trace formula is related to either a state formula or a normalized trace formula, which can be found in linear time.

Lemma 2 (Soundness of Normalization)**.** *If $\pi \rightsquigarrow \xi$ then for all traces σ, $\sigma \vDash \pi$ if and only if $\sigma \vDash \xi$.*

Proof. Soundness of $\rightsquigarrow\phi$ is trivial. Soundness of proof rules $\rightsquigarrow\sqcap$, $\rightsquigarrow\sqcup$, $\rightsquigarrow\blacktriangleleft\Box$ and $\rightsquigarrow\blacktriangleleft\Diamond$ is true by Def. 3, keeping in mind the intuition given in Remark 1. Soundness of proof rule $\sim\rightsquigarrow$ is by induction and using Lemma 1. □

Lemma 3 (Existence of a Normalized Form)**.** *For any trace formula π there exists a state formula ϕ such that $\pi \rightsquigarrow \phi$, or a normalized trace formula ξ such that $\pi \rightsquigarrow \xi$. Such a ϕ or ξ can be computed from π in linear time.*

Proof. This lemma is a direct consequence of Lemma 1, using the identities of Fig. 1. Unless π is itself a state formula ϕ, it is related to a normalized trace formula ξ. □

Lemma 3 concludes our study of normalized forms. Since every trace formula is related (and thus semantically equivalent by Lemma 2) to a state formula or a normalized trace formula, we can limit our axiomatization to the study of state formulas and normalized trace formulas. Formulas of the form $[\alpha]\phi$ or $\langle\alpha\rangle\phi$ involving state formulas have already been axiomatized in $\mathsf{d}\mathcal{L}$ [Pla08,

Pla12], [Pla10, chapter 2] (for reference we repeat this axiomatization in Appendix D). The rest of this paper focuses on axiomatizing formulas of the form $[\alpha]\xi$ or $\langle\alpha\rangle\xi$ involving normalized trace formulas. In Appendix A.1, we come back to trace formulas to study a direct treatment of proof rules for state formulas of the form $[\alpha]\pi$ and $\langle\alpha\rangle\pi$ in order to make the system more efficient.

3.3 Proof Calculus for dTL^2

In this section we present a proof calculus for dTL^2 for verifying temporal properties of hybrid programs specified in the differential temporal dynamic logic dTL^2. The basic idea of the proof calculus is symbolic decomposition. The calculus progressively transforms formulas to simpler formulas, often by inductively decomposing programs that are in program modalities. In particular, the temporal rules progressively transform temporal formulas to temporal-free formulas, in order to leverage the nontemporal rules of $\mathsf{d}\mathcal{L}$. The proof system inherits its nontemporal rules from the $\mathsf{d}\mathcal{L}$ proof system [Pla08, Pla12], [Pla10, chapter 2], and adds its own temporal rules. As is the case for $\mathsf{d}\mathcal{L}$, the basis of our proof system is real arithmetic, and we integrate it as in $\mathsf{d}\mathcal{L}$ [Pla08, Pla12], [Pla10, chapter 2]. We first present how to use the rules, then a brief overview on the inherited nontemporal rules from $\mathsf{d}\mathcal{L}$, and finally a detailed account of the new temporal rules of dTL^2, summarized in Fig. 2.

Usage of the Rules. Rules are to be used in the same way as in the $\mathsf{d}\mathcal{L}$ calculus. We do, however, use a new double bar notation by writing some rules in the form

$$\frac{\phi}{\psi}$$

This notation denotes equivalence of the premise and its conclusion. This means that there exists a dual rule, hence the two following rules are true

$$\frac{\phi}{\psi} \qquad\qquad \frac{\neg\phi}{\neg\psi}$$

For space reasons we do not list dual rules explicitly but give them in Appendix C.

Inherited Nontemporal Rules. On top of the temporal rules presented in Fig. 2, the proof calculus of dTL^2 also inherits the rules of the proof calculus of $\mathsf{d}\mathcal{L}$. Since the semantics of dTL^2 conservatively extends the semantics of dTL, which itself conservatively extends the semantics of $\mathsf{d}\mathcal{L}$ [Pla07], [Pla10, chapter 4], it is sound to inherit the $\mathsf{d}\mathcal{L}$ calculus. While we inherit the nontemporal rules of $\mathsf{d}\mathcal{L}$, we do not inherit — but rather reformulate with normalized trace formulas — the temporal rules of dTL [Pla07], [Pla10, chapter 4], thus enabling more efficient proofs by exploiting normalized trace formulas.

Temporal Rules. The temporal rules of the proof calculus of dTL^2 are presented in Fig. 2, in which they are grouped by program construct. Rules $[\,]\rightsquigarrow$ and $\langle\,\rangle\rightsquigarrow$ lift trace formula normalization to program modalities. Rule $[\cup]\xi$ for nondeterministic choice easily extends corresponding

Normalization of Trace Formulas

$$\frac{\pi \rightsquigarrow \xi \quad [\alpha]\xi}{[\alpha]\pi}([\,]\rightsquigarrow) \qquad \frac{\pi \rightsquigarrow \xi \quad \langle\alpha\rangle\xi}{\langle\alpha\rangle\pi}(\langle\,\rangle\rightsquigarrow)$$

Sequential Composition

$$\frac{[\alpha]([\beta](\phi \sqcap \Box\psi) \sqcap \Box\psi)}{[\alpha;\beta](\phi \sqcap \Box\psi)}([;]\sqcap) \qquad \frac{[\alpha]([\beta](\phi \sqcup \Diamond\psi) \sqcup \Diamond\psi)}{[\alpha;\beta](\phi \sqcup \Diamond\psi)}([;]\sqcup) \qquad \frac{[\alpha]([\beta](\phi \blacktriangleleft \pi) \blacktriangleleft \pi)}{[\alpha;\beta](\phi \blacktriangleleft \pi)}([;]\blacktriangleleft)$$

Nondeterministic Choice

$$\frac{[\alpha]\xi \wedge [\beta]\xi}{[\alpha \cup \beta]\xi}([\cup]\xi)$$

Test

$$\frac{(\neg\chi \vee \phi) \wedge \psi}{[?\chi](\phi \sqcap \Box\psi)}([?]\sqcap) \qquad \frac{(\chi \wedge \phi) \vee (\neg\chi \wedge \psi)}{[?\chi](\phi \blacktriangleleft \Diamond\Box\psi)}([?]\blacktriangleleft\Diamond)$$

$$\frac{(\chi \wedge \phi) \vee \psi}{[?\chi](\phi \sqcup \Diamond\psi)}([?]\sqcup) \qquad \frac{(\chi \wedge \phi) \vee (\neg\chi \wedge \psi)}{[?\chi](\phi \blacktriangleleft \Box\Diamond\psi)}([?]\blacktriangleleft\Box)$$

Assignment

$$\frac{\psi \wedge [x := \theta](\phi \wedge \psi)}{[x := \theta](\phi \sqcap \Box\psi)}([:=]\sqcap) \qquad \frac{\psi \vee [x := \theta](\phi \vee \psi)}{[x := \theta](\phi \sqcup \Diamond\psi)}([:=]\sqcup) \qquad \frac{[x := \theta]\phi}{[x := \theta](\phi \blacktriangleleft \pi)}([:=]\blacktriangleleft)$$

Ordinary Differential Equation

$$\frac{\psi \wedge [x' = \theta \,\&\, \chi](\phi \wedge \psi)}{[x' = \theta \,\&\, \chi](\phi \sqcap \Box\psi)}([']\sqcap)$$

$$\frac{(\chi \vee \psi) \wedge [x' = \theta \,\&\, (\chi \wedge \neg\psi)]\phi \wedge \langle x' = \theta\rangle(\neg\chi \vee \psi)}{[x' = \theta \,\&\, \chi](\phi \sqcup \Diamond\psi)}([']\sqcup)$$

$$\frac{(\chi \vee \psi) \wedge [x' = \theta \,\&\, \chi]\phi \wedge (\langle x' = \theta\rangle(\neg\chi) \vee \langle x' = \theta\rangle[x' = \theta]\psi)}{[x' = \theta \,\&\, \chi](\phi \blacktriangleleft \Diamond\Box\psi)}([']\blacktriangleleft\Diamond)$$

$$\frac{(\chi \vee \psi) \wedge [x' = \theta \,\&\, \chi]\phi \wedge (\langle x' = \theta\rangle(\neg\chi) \vee [x' = \theta]\langle x' = \theta\rangle\psi)}{[x' = \theta \,\&\, \chi](\phi \blacktriangleleft \Box\Diamond\psi)}([']\blacktriangleleft\Box)$$

Repetition

$$\frac{\phi \wedge [\alpha^*][\alpha](\phi \sqcap \Box\psi)}{[\alpha^*](\phi \sqcap \Box\psi)}([^*]\sqcap) \qquad \frac{\psi \vee (\phi \wedge [\alpha;\alpha^*](\phi \sqcup \Diamond\psi))}{[\alpha^*](\phi \sqcup \Diamond\psi)}([^{*n}]\sqcup)$$

$$\frac{\forall^\alpha(\phi \to [\alpha](\phi \sqcup \Diamond\psi))}{\phi \to [\alpha^*](\phi \sqcup \Diamond\psi)}(\mathsf{ind}\sqcup) \qquad \frac{\phi \wedge [\alpha^*][\alpha](\phi \blacktriangleleft \pi)}{[\alpha^*](\phi \blacktriangleleft \pi)}([^*]\blacktriangleleft)$$

$$\frac{\forall^\alpha \forall r > 0\ (\varphi(r) \to \langle\alpha\rangle(\varphi(r-1) \sqcap \Box\psi))}{(\exists r\ \varphi(r)) \wedge \psi \to \langle\alpha^*\rangle((\exists r \leq 0\ \varphi(r)) \sqcap \Box\psi)}(\mathsf{con}\sqcap)$$

Figure 2: Rule schemata of the proof calculus for dTL^2

rule $[\cup]$ of $\mathsf{d}\mathcal{L}$, and assignment rules behave as expected, largely because assignments always terminate.

The sequential composition rules exhibit how nicely the normalized formula interact with sequential composition; remember that sequential composition is one of the main technical difficulties of a calculus handling alternating program and temporal modalities. Normalized trace formulas were designed for these rules, and particular care was taken in considering nonterminating traces. Rule $[;]\sqcap$ expresses that all traces of the composition of two programs α and β satisfies $\phi \sqcap \Box\psi$ if and only if all traces of α satisfy $\Box\psi$, and for terminating traces of α, if all following traces of β satisfy $\phi \sqcap \Box\psi$. In particular, this rule improves on the corresponding rule $[;]\Box$ of dTL by *not* duplicating program modality $[\beta]$, thus eliminating proofs that are exponential in the number of sequential compositions. Rule $[;]\sqcup$ is the main rule for alternating program and temporal modalities in the context of sequential composition. It expresses that all traces of the composition of two programs α and β satisfies $\phi \sqcap \Diamond\psi$ if and only if all traces of α either satisfy $\Diamond\psi$, or are terminating and followed only by traces of β satisfying $\phi \sqcup \Diamond\psi$. Finally, rule $[;]\blacktriangleleft$ similarly handles sequential compositions followed by a $\blacktriangleleft$ operator.

For the test rules, let us remember that a test trace terminates only if the test passes, and is otherwise an error trace. Any trace of test $?\chi$ satisfies $\phi \sqcap \Box\psi$ if and only if its initial state satisfies $\phi \wedge \psi$ when it terminates, or satisfies just ψ when it doesn't terminate; this can be summarized as $(\neg\chi \vee \phi) \wedge \psi$ as in rule $[?]\sqcap$. Rule $[?]\sqcup$ is similar. Any trace of test $?\chi$ satisfies $\phi \blacktriangleleft \Diamond\Box\psi$ if and only if it terminates and its initial state satisfied ϕ, or it doesn't terminate and its initial state satisfied ψ; this can be summarized as $(\chi \wedge \phi) \vee (\neg\chi \wedge \psi)$ as in rule $[?] \blacktriangleleft \Diamond$. Rule $[?] \blacktriangleleft \Box$ is similar.

Ordinary differential equations have terminating traces, but also infinite and error traces. Additionally, the execution can exit a differential equation at any moment, even if the evolution constraint domain it still verified; thus formulas like $[x' = \theta \,\&\, \chi]\phi$ and $[x' = \theta \,\&\, \chi]\Box\phi$ are equivalent in a state satisfying χ. Rules for ordinary differential equations transform formulas into temporal-free formulas, on which the $\mathsf{d}\mathcal{L}$ proof calculus and in particular differential invariants can be used. In rule $[']\sqcap$, the first conjunct ψ is necessary to handle error traces, when χ is initially false. In rule $[']\sqcup$, the first conjunct $\chi \vee \psi$ expresses that the differential equation can evolve or has satisfied $\Diamond\psi$ initially. The second conjunct handles traces that never satisfy ψ and thus have to satisfy ϕ, and the third conjunct makes sure there is either no infinite trace ($\langle x' = \theta\rangle\neg\chi$), or that such an infinite trace satisfies $\Diamond\psi$ (condition $\langle x' = \theta\rangle\psi$, equivalent to $\langle x' = \theta\rangle\Diamond\psi$). The first conjunct of rule $['] \blacktriangleleft \Diamond$ again handles error traces as in rule $[']\sqcup$. The second conjunct ensures all terminating traces finish in a state satisfying ϕ, and its third conjunct handles infinite traces by making sure they don't exist ($\langle x' = \theta\rangle\neg\chi$) or that they satisfy $\Diamond\Box\psi$ (condition $\langle x' = \theta\rangle[x' = \theta]\psi$). Rule $['] \blacktriangleleft \Box$ is similar.

In some way, repetition rules are easier because as long as a repetition only repeats a terminating trace, it is itself terminating. Rules $[^*]\sqcap$ and $[^*]\blacktriangleleft$ are particularly satisfying because their premise no longer contains a temporal property of a loop, but only a non-temporal postcondition of a loop, which is thus provable by ordinary, non-temporal induction. Only the postcondition still has a temporal property but no more loops. That is, these rules reduce temporal properties of loops to nontemporal properties of loops, or more complicated temporal properties on a program without the loop. In rule $[^*]\sqcap$, the first disjunct expresses that $\Diamond\psi$ holds without repeating if ψ holds

initially. The first conjunct ϕ of the second disjunct is necessary when α repeats zero times; while the second conjunct executes α any number of times n, then checks that the $(n+1)$-st execution of α also satisfies $\phi \sqcap \Box\psi$. The treatment of rule $[^*]\blacktriangleleft$ is similar. Rule $[^{*n}]$ is less satisfying because it leaves an α^* inside a program modality followed by a normalized trace formula. If ψ is true then the conclusion trivially holds; otherwise the rule relies on the fact that α^* is equivalent to $?\mathsf{true} \cup \alpha;\alpha^*$ and just unwinds the loop once. Program $\alpha;\alpha^*$ in the modality could as well be the equivalent $\alpha^*;\alpha$. The same thing is *not* true for rule $[^*]\sqcap$, where $[\alpha^*][\alpha](\phi \sqcap \Box\phi)$ ensures progress of the proof, while writing $[\alpha][\alpha^*](\phi \sqcap \Box\phi)$ would not. Rules $\mathsf{ind}\sqcup$ and $\mathsf{con}\sqcap$ extend induction (ind) and convergence (con) rules of $\mathsf{d}\mathcal{L}$ to normalized trace formulas. As in $\mathsf{d}\mathcal{L}$, they are not equivalences; and also as in $\mathsf{d}\mathcal{L}$, they use the notation $\forall^\alpha$, which quantifies over all variables possibly assigned by α in assignments or differential equations. Rule $\mathsf{ind}\sqcup$ shows that ϕ is inductive with exit clause $\Diamond\psi$, i.e., ϕ holds after all traces of α from any state where ϕ holds, except when exit condition ψ was true at some point during that trace. If ψ was true initially, rule $[^{*n}]$ applies instead. Rule $\mathsf{con}\sqcap$ proves that φ is a variant of some trace of α (i.e., its level r decreases) during which ψ always holds true. Then starting from some initial r (assumption of conclusion), an r for which $\varphi(r)$ holds will ultimately be ≤ 0 without having violated when repeating α^* often enough.

3.4 Meta-Results

Soundness. The following result shows that verification with the dTL^2 calculus always produces correct results about the temporal behavior of hybrid systems, i.e., the dTL^2 calculus presented in Fig. 2 is sound. Theorem 4 is proved in Appendix B.2.

Theorem 4 (Soundness of dTL^2)**.** *The dTL^2 calculus presented in Fig. 2 is sound, i.e., derivable state formulas are valid, i.e., valid in every state.*

Incompleteness of dTL^2. In [Pla08, Pla12], [Pla10, chapter 2] it was shown that the discrete and continuous fragments of $\mathsf{d}\mathcal{L}$ are non-axiomatizable. An extension of $\mathsf{d}\mathcal{L}$, the logic dTL is also non-axiomatizable [Pla07], [Pla10, chapter 4]. Since dTL^2 is a conservative extension of both $\mathsf{d}\mathcal{L}$ and dTL, those results lift to dTL^2. Therefore the discrete and continuous fragments of dTL^2, even if only containing nontemporal formulas are non-axiomatizable. In particular dTL^2 is non-axiomatizable.

Relative Completeness for Star-Free Expressions. We now show how to lift the relative completeness result of $\mathsf{d}\mathcal{L}$ [Pla08, Pla12], [Pla10, chapter 2] to dTL^2; this completeness result is relative to first order logic of differential equations (FOD), i.e., first-order real arithmetic augmented with formulas expressing properties of differential equations [Pla08, Pla12], [Pla10, chapter 2].

Theorem 5 (Relative completeness for star-free expressions)**.** *The dTL^2 calculus restricted to *-free programs is complete relative to FOD, i.e., every valid dTL^2 formula with only star-free programs can be derived from FOD tautologies.*

Theorem 5 is proved in Appendix B.3. We conjecture that the proof system of dTL^2 is also relatively complete relative to FOD for all expressions, including repetitions.

4 Alternative Proof Systems

Normalizing all temporal formulas before applying the rules of Fig. 2 can sometimes result in longer proofs than necessary. In Appendix A.1 we study a proof system directly handling (non-normalized) trace formulas. This extended proof system alleviates the need for normalizing all trace formulas, and is thus more efficient.

Another alternative, that we also study in Appendix A.2, is to suppress all the $[\,]\sqcap$ rules of Fig. 2 (rules $[;]\sqcap$, $[?]\sqcap$, $[:=]\sqcap$, $[']\sqcap$ and $[^*]\sqcap$) and replace them by rules directly handling formulas of the form $[\alpha]\Box\phi$, and the following rule:

$$\frac{[\alpha]\phi \wedge [\alpha]\Box\psi}{[\alpha](\phi \sqcap \Box\psi)}([\,]\sqcap)$$

This results in a simpler system, because some of the rules are less complicated. However the system is not as efficient, because it duplicates the symbolic execution of α.

5 Related Work

In this section we study work related specifically to temporal reasoning of hybrid systems. For a more general account of previous work on verification of hybrid systems we refer to [Pla08, Pla12], [Pla10, chapter 2].

This paper is based on work by Platzer introducing a temporal dynamic logic for hybrid systems [Pla07], extending previous work by Beckert and Schlager [BS01] to hybrid programs. Both papers present a relatively complete calculus; however Beckert and Schlager only consider discrete state spaces, and only study temporal formulas of the form $[\alpha]\Box\phi$ and its dual $\langle\alpha\rangle\Diamond\phi$, leaving out any mixed cases alternating program and temporal modalities $[\alpha]\Diamond\phi$ or $[\alpha]\Box\Diamond\phi$. Platzer proposes to handle mixed cases by nonlocal program transformation, but does not show how to handle them compositionally.

Process logic [HKP82, Nis80, Par78, Pra79] initially used temporal logic [EH86, Pnu77] in the context of dynamic logic [HKT00] to reason about temporal behavior of programs. It is well studied, but limited to discrete programs. It also only considers an abstract notion of atomic program, without explicitly considering assignments and tests.

Davoren and Nerode [DN00] study hybrid systems and their topological aspects in the context of the propositional modal μ-calculus. Davoren, Coulthard, Markey and Moor [DCMM04] also give a semantics in general flow systems for a generalization of CTL*. In both [DN00] and [DCMM04], the authors provide Hilbert-style calculi to prove formulas of their systems, but in a propositional — not first-order — system, without specific proof rules to handle ordinary differential equations. Zhou, Ravn and Hansen [ZRH92] present a duration calculus extended by mathematical expressions with derivatives of state variables. Their system requires external mathematical reasoning about derivatives and continuity.

Other authors have studied temporal properties of hybrid systems in the context of model checking. Mysore, Piazza and Mishra [MPM05] study model checking of semi-algebraic hybrid systems for TCTL (Timed Computation Tree Logic) properties and prove undecidability. They

do bounded model checking for differential equations with polynomial solutions only, while we handle more general polynomial differential equations and unbounded safety verification. Additionally TCTL does not allow nesting of temporal modalities as we do. Cimatti, Roveri and Tonetta [CRT09] present HRELTL, a linear temporal logic with regular expressions for hybrid traces. Their work is inspired by requirements validation for the European Train Control System, and uses bounded model checking and satisfiability modulo theory. More recently, Bresolin [Bre13] develops HyLTL, a temporal logic for model checking hybrid systems, and shows how to solve the model checking problem by translating formulas into equivalent hybrid automata.

6 Conclusion and Future Work

In this paper we have presented a proof calculus for dTL^2, extending dTL by allowing nesting of temporal modalities. We showed proof rules for handling compositionally alternating program and temporal modalities, solving an open problem formulated in 2001 [BS01] and identified as a problem for hybrid systems in 2007 [Pla07], [Pla10, chapter 4]. We also offered a treatment where programs are not duplicated by proof rules, solving another open problem formulated by [Pla07], [Pla10, chapter 4]. We showed that the system is relatively complete with respect to FOD for *-free hybrid programs. The treatment of infinite traces is crucial to make the logic interesting, as temporal properties on terminating and error traces simplify greatly (Remark 1).

Future work includes proving our conjecture that the system is relatively complete with respect to FOD for all expressions; extending the semantics and the proof system to allow repetition — and not just differential equations — to create infinite traces; and implementing our proof rules in a tool such as KeYmaera [PQ08].

A number of extensions to dTL^2 should be explored, such as inclusion of the temporal Until operator, or nested conjunctions and disjunctions inside temporal formulas. Some of these extensions can be handled by program transformations [Pla07], [Pla10, chapter 4], but a compositional proof system such as the one presented here would be more interesting. The proof system of dTL^2 is an important step towards a more general system dTL*, extending dTL^2 with formulas of CTL*, and expressing formulas such as $[\alpha]\Box(\Diamond\phi \wedge \psi)$. We would like to develop a semantics and a proof system for dTL*.

Acknowledgements. We are grateful to Khalil Ghorbal, Dexter Kozen, Sarah Loos, Stefan Mitsch, Ed Morehouse, Jan-David Quesel, Marcus Völp, and the anonymous referees for helpful comments and discussions.

References

[Bre13] Davide Bresolin. HyLTL: a temporal logic for model checking hybrid systems. In Luca Bortolussi, Manuela L. Bujorianu, and Giordano Pola, editors, *HAS*, volume 124 of *EPTCS*, pages 73–84, 2013. doi:10.4204/EPTCS.124.8.

[BS01] Bernhard Beckert and Steffen Schlager. A sequent calculus for first-order dynamic logic with trace modalities. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *IJCAR*, volume 2083 of *LNCS*, pages 626–641. Springer, 2001. doi:10.1007/3-540-45744-5_51.

[CRT09] Alessandro Cimatti, Marco Roveri, and Stefano Tonetta. Requirements validation for hybrid systems. In Ahmed Bouajjani and Oded Maler, editors, *CAV*, volume 5643 of *LNCS*, pages 188–203. Springer, 2009. doi:10.1007/978-3-642-02658-4_17.

[DCMM04] Jennifer M. Davoren, Vaughan Coulthard, Nicolas Markey, and Thomas Moor. Non-deterministic temporal logics for general flow systems. In Rajeev Alur and George J. Pappas, editors, *HSCC*, volume 2993 of *LNCS*, pages 280–295. Springer, 2004. doi:10.1007/978-3-540-24743-2_19.

[DN00] Jennifer M. Davoren and Anil Nerode. Logics for hybrid systems. *Proceedings of the IEEE*, 88(7):985–1010, July 2000. doi:10.1109/5.871305.

[EH86] E. Allen Emerson and Joseph Y. Halpern. "Sometimes" and "Not Never" revisited: on branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, 1986. doi:10.1145/4904.4999.

[HC96] George E. Hughes and Max J. Cresswell. *A New Introduction to Modal Logic*. Routledge, 1996.

[HKP82] David Harel, Dexter Kozen, and Rohit Parikh. Process logic: Expressiveness, decidability, completeness. *J. Comput. Syst. Sci.*, 25(2):144–170, 1982. doi:10.1016/0022-0000(82)90003-4.

[HKT00] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*. MIT Press, Cambridge, MA, 2000.

[MPM05] Venkatesh Mysore, Carla Piazza, and Bud Mishra. Algorithmic algebraic model checking II: Decidability of semi-algebraic model checking and its applications to systems biology. In Doron Peled and Yih-Kuen Tsay, editors, *ATVA*, volume 3707 of *LNCS*, pages 217–233. Springer, 2005. doi:10.1007/11562948_18.

[Nis80] Hirokazu Nishimura. Descriptively complete process logic. *Acta Inf.*, 14:359–369, 1980. doi:10.1007/BF00286492.

[Par78] Rohit Parikh. A decidability result for a second order process logic. In *FOCS*, pages 177–183. IEEE Comp. Soc., 1978. doi:10.1109/SFCS.1978.2.

[Pla07] André Platzer. A temporal dynamic logic for verifying hybrid system invariants. In Sergei N. Artëmov and Anil Nerode, editors, *LFCS*, volume 4514 of *LNCS*, pages 457–471. Springer, 2007. doi:10.1007/978-3-540-72734-7_32.

[Pla08] André Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reas.*, 41(2):143–189, 2008. doi:10.1007/s10817-008-9103-8.

[Pla10] André Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer, Heidelberg, 2010. doi:10.1007/978-3-642-14509-4.

[Pla12] André Platzer. Logics of dynamical systems. In *LICS*, pages 13–24. IEEE, 2012. doi:10.1109/LICS.2012.13.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57. IEEE Comp. Soc., 1977. doi:10.1109/SFCS.1977.32.

[PQ08] André Platzer and Jan-David Quesel. KeYmaera: A hybrid theorem prover for hybrid systems. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *IJCAR*, volume 5195 of *LNCS*, pages 171–178. Springer, 2008. doi:10.1007/978-3-540-71070-7_15.

[Pra79] Vaughan R. Pratt. Process logic. In Alfred V. Aho, Stephen N. Zilles, and Barry K. Rosen, editors, *POPL*, pages 93–100. ACM, 1979. doi:10.1145/567752.567761.

[ZRH92] Chaochen Zhou, Anders P. Ravn, and Michael R. Hansen. An extended duration calculus for hybrid real-time systems. In Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors, *Hybrid Systems*, volume 736 of *LNCS*, pages 36–59. Springer, 1992. doi:10.1007/3-540-57318-6_23.

A Alternative Proof Systems

A.1 Direct proofs of (non-normalized) trace formulas

Normalizing all temporal formulas before applying the rules of Fig. 2 can sometimes result in longer proofs than necessary. In Fig. 3 we show a proof system directly handling (non-normalized) trace formulas. These rules do not replace the rules of Fig. 2, but are rather added to them. However using the rules of Fig. 3 alleviates the need for normalizing all trace formulas, and thus the definition of the relation $\rightsquigarrow$ can be dropped (but not the definition of normalized trace formulas), without losing any proving power.

In this new proof system, we use Lemma 1 to transform any trace formula into an equivalent trace formula of the form ϕ, $\Box\phi$, $\Diamond\phi$, $\Box\Diamond\phi$ or $\Diamond\Box\phi$. Rules $[\,]\sim$ and $\langle\,\rangle\sim$ lift this transformation to program modalities. However, we do not transform it into a normalized trace formula. Most of the rules in Fig. 3 correspond to a closely related rule in Fig. 2, and are obtained from it by replacing formulas $\Box\phi$, $\Diamond\phi$, $\Box\Diamond\phi$ and $\Diamond\Box\phi$ by their normalized equivalent, and simplifying.

Noteworthy are the rules for sequential compositions, because their premises make use of normalized trace formulas, while their conclusions do not; these rules indicate that *normalized trace formulas cannot be avoided* and are *naturally introduced*. Rules for ordinary differential equations simplify greatly compared to Fig. 2, as some premises become unnecessary because implied by other premises. Finally, rule $[^*]\Diamond$ is particularly interesting: it is not a trivial consequence of any single rule of Fig. 2, but it can be proved from $\mathsf{ind}\sqcup$ and $\mathsf{con}\sqcap$; it is however simpler to prove it directly. It also shows a more disappointing fact: $[\alpha^*]\Diamond\phi$ is equivalent to ϕ, and therefore it cannot express any interesting property.

Soundness of the rules of Fig. 3

Proof. Most of the rules of Fig. 3 are direct consequences of a rule of Fig. 2, using Lemma 2 on rules of Fig. 1. The only exception — and a slightly more difficult case — is the rule $[^*]\Diamond$. We now treat this case in detail.

$[^*]\Diamond$ The soundness of $[^*]\Diamond$ is a consequence of rule $\mathsf{ind}\sqcup$ of Fig. 2, while the soundness of its converse is a consequence of rule $\mathsf{con}\sqcap$ of Fig. 2. If $v \vDash \phi$, then $v \vDash \mathsf{false} \vee \phi$, and $v \vDash \forall^\alpha(\mathsf{false} \to [\alpha](\mathsf{false} \sqcup \Diamond\phi))$ vacuously. Therefore we can apply rule $\mathsf{ind}\sqcup$ to conclude that $v \vDash [\alpha^*](\mathsf{false} \sqcup \Diamond\phi)$, which is the same as $v \vDash [\alpha^*]\Diamond\phi$. For the converse, let us prove the dual rule instead:

$$\frac{\phi}{\langle\alpha^*\rangle\Box\phi}(\langle^*\rangle\Box)$$

If $v \vDash \phi$, then let us define the real function φ as $\varphi(r) = \mathsf{false}$ if $r > 0$ and $\varphi(r) = \mathsf{true}$ if $r \leq 0$. Then $v \vDash (\exists r\ \varphi(r)) \wedge \phi$, and vacuously $v \vDash \forall^\alpha \forall r > 0\ (\varphi(r) \to \langle\alpha\rangle(\varphi(r-1) \sqcap \Box\phi))$, since $\varphi(r) = \mathsf{false}$ for $r > 0$. Therefore we can apply rule $\mathsf{con}\sqcap$ to conclude that $\langle\alpha^*\rangle((\exists r \leq 0\ \varphi(r)) \sqcap \Box\psi)$, which is the same as $\langle\alpha^*\rangle\Box\psi$.

Equivalence of Trace Formulas

$$\frac{\pi_1 \sim \pi_2 \quad [\alpha]\pi_2}{[\alpha]\pi_1}([\,]\sim) \qquad \frac{\pi_1 \sim \pi_2 \quad \langle\alpha\rangle\pi_2}{\langle\alpha\rangle\pi_1}(\langle\,\rangle\sim)$$

Sequential Composition

$$\frac{[\alpha]([\beta]\Box\phi \sqcap \Box\phi)}{[\alpha;\beta]\Box\phi}([;]\Box) \qquad \frac{[\alpha]([\beta]\Diamond\phi \sqcup \Diamond\phi)}{[\alpha;\beta]\Diamond\phi}([;]\Diamond)$$

$$\frac{[\alpha]([\beta]\Diamond\Box\phi \blacktriangleleft \Diamond\Box\phi)}{[\alpha;\beta]\Diamond\Box\phi}([;]\Diamond\Box) \qquad \frac{[\alpha]([\beta]\Box\Diamond\phi \blacktriangleleft \Box\Diamond\phi)}{[\alpha;\beta]\Box\Diamond\phi}([;]\Box\Diamond)$$

Nondeterministic Choice

$$\frac{[\alpha]\pi \wedge [\beta]\pi}{[\alpha \cup \beta]\pi}([\cup]\pi)$$

Test

$$\frac{\phi}{[?\chi]\Box\phi}([?]\Box) \qquad \frac{\phi}{[?\chi]\Diamond\phi}([?]\Diamond) \qquad \frac{\phi}{[?\chi]\Diamond\Box\phi}([?]\Diamond\Box) \qquad \frac{\phi}{[?\chi]\Box\Diamond\phi}([?]\Box\Diamond)$$

Assignment

$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}([:=]\Box) \qquad \frac{\phi \vee [x := \theta]\phi}{[x := \theta]\Diamond\phi}([:=]\Diamond)$$

$$\frac{[x := \theta]\phi}{[x := \theta]\Diamond\Box\phi}([:=]\Diamond\Box) \qquad \frac{[x := \theta]\phi}{[x := \theta]\Box\Diamond\phi}([:=]\Box\Diamond)$$

Differential Equation

$$\frac{\phi \wedge [x' = \theta \,\&\, \chi]\phi}{[x' = \theta \,\&\, \chi]\Box\phi}([']\Box) \qquad \frac{\phi}{[x' = \theta \,\&\, \chi]\Diamond\phi}([']\Diamond)$$

$$\frac{(\chi \vee \phi) \wedge [x' = \theta \,\&\, \chi]\phi}{[x' = \theta \,\&\, \chi]\Diamond\Box\phi}([']\Diamond\Box) \qquad \frac{(\chi \vee \phi) \wedge [x' = \theta \,\&\, \chi]\phi}{[x' = \theta \,\&\, \chi]\Box\Diamond\phi}([']\Box\Diamond)$$

Repetition

$$\frac{[\alpha^*][\alpha]\Box\phi}{[\alpha^*]\Box\phi}([^*]\Box) \qquad \frac{\phi}{[\alpha^*]\Diamond\phi}([^*]\Diamond) \qquad \frac{\phi \wedge [\alpha^*][\alpha]\Diamond\Box\phi}{[\alpha^*]\Diamond\Box\phi}([^*]\Diamond\Box) \qquad \frac{\phi \wedge [\alpha^*][\alpha]\Box\Diamond\phi}{[\alpha^*]\Box\Diamond\phi}([^*]\Box\Diamond)$$

Figure 3: Rules of the proof calculus for direct proofs of (non-normalized) trace formulas

$[\,]\sim$ Soundness of $[\,]\sim$ is a direct consequence of rule $[\,]\leadsto$, using Lemmas 2 and 3 on π_1.

$\langle\,\rangle\sim$ Soundness of $\langle\,\rangle\sim$ is a direct consequence of rule $\langle\,\rangle\leadsto$, using Lemmas 2 and 3 on π_1.

$[;]\Box$ Soundness of $[;]\Box$ and its converse are a direct consequence of rule $[;]\sqcap$ and its converse, using Lemma 2 on $\Box\phi \leadsto \mathsf{true} \sqcap \Box\phi$ by rule $\leadsto\sqcap$.

$[;]\Diamond$ Soundness of $[;]\Diamond$ and its converse are a direct consequence of rule $[;]\sqcup$ and its converse, using Lemma 2 on $\Diamond\phi \leadsto \mathsf{false} \sqcup \Diamond\phi$ by rule $\leadsto\sqcup$.

$[;]\Diamond\Box$ Soundness of $[;]\Diamond\Box$ and its converse are a direct consequence of rule $[;]\blacktriangleleft$ and its converse, using Lemma 2 on $\Diamond\Box\phi \leadsto \phi \blacktriangleleft \Diamond\Box\phi$ by rule $\leadsto\blacktriangleleft\Diamond$.

$[;]\Box\Diamond$ Soundness of $[;]\Box\Diamond$ and its converse are a direct consequence of rule $[;]\blacktriangleleft$ and its converse, using Lemma 2 on $\Box\Diamond\phi \leadsto \phi \blacktriangleleft \Box\Diamond\phi$ by rule $\leadsto\blacktriangleleft\Box$.

$[\cup]\pi$ Soundness of $[\cup]\pi$ and its converse are a direct consequence of rule $[\cup]\xi$ and its converse, using Lemmas 2 and 3 on π.

$[?]\Box$ Soundness of $[?]\Box$ and its converse are a direct consequence of rule $[?]\sqcap$ and its converse, using Lemma 2 on $\Box\phi \leadsto \mathsf{true} \sqcap \Box\phi$ by rule $\leadsto\sqcap$.

$[?]\Diamond$ Soundness of $[?]\Diamond$ and its converse are a direct consequence of rule $[?]\sqcup$ and its converse, using Lemma 2 on $\Diamond\phi \leadsto \mathsf{false} \sqcup \Diamond\phi$ by rule $\leadsto\sqcup$.

$[?]\Diamond\Box$ Soundness of $[?]\Diamond\Box$ and its converse are a direct consequence of rule $[?] \blacktriangleleft \Diamond$ and its converse, using Lemma 2 on $\Diamond\Box\phi \leadsto \phi \blacktriangleleft \Diamond\Box\phi$ by rule $\leadsto\blacktriangleleft\Diamond$.

$[?]\Box\Diamond$ Soundness of $[?]\Box\Diamond$ and its converse are a direct consequence of rule $[?] \blacktriangleleft \Box$ and its converse, using Lemma 2 on $\Box\Diamond\phi \leadsto \phi \blacktriangleleft \Box\Diamond\phi$ by rule $\leadsto\blacktriangleleft\Box$.

$[:=]\Box$ Soundness of $[:=]\Box$ and its converse are a direct consequence of rule $[:=]\sqcap$ and its converse, using Lemma 2 on $\Box\phi \leadsto \mathsf{true} \sqcap \Box\phi$ by rule $\leadsto\sqcap$.

$[:=]\Diamond$ Soundness of $[:=]\Diamond$ and its converse are a direct consequence of rule $[:=]\sqcup$ and its converse, using Lemma 2 on $\Diamond\phi \leadsto \mathsf{false} \sqcup \Diamond\phi$ by rule $\leadsto\sqcup$.

$[:=]\Diamond\Box$ Soundness of $[:=]\Diamond\Box$ and its converse are a direct consequence of rule $[:=] \blacktriangleleft \Diamond$ and its converse, using Lemma 2 on $\Diamond\Box\phi \leadsto \phi \blacktriangleleft \Diamond\Box\phi$ by rule $\leadsto\blacktriangleleft\Diamond$.

$[:=]\Box\Diamond$ Soundness of $[:=]\Box\Diamond$ and its converse are a direct consequence of rule $[:=] \blacktriangleleft \Box$ and its converse, using Lemma 2 on $\Box\Diamond\phi \leadsto \phi \blacktriangleleft \Box\Diamond\phi$ by rule $\leadsto\blacktriangleleft\Box$.

$[']\Box$ Soundness of $[']\Box$ and its converse are a direct consequence of rule $[']\sqcap$ and its converse, using Lemma 2 on $\Box\phi \leadsto \mathsf{true} \sqcap \Box\phi$ by rule $\leadsto\sqcap$.

$[']\Diamond$ Soundness of $[']\Diamond$ and its converse are a consequence of rule $[']\sqcup$ and its converse, using Lemma 2 on $\Diamond\phi \leadsto \mathsf{false} \sqcup \Diamond\phi$ by rule $\leadsto\sqcup$, and noticing that $v \vDash [x' = \theta \,\&\, (\chi \wedge \neg\phi)]\mathsf{true} \wedge \langle x' = \theta\rangle(\neg\chi \vee \phi)$ is implied by $v \vDash \phi$.

$[']\Diamond\Box$ Soundness of $[']\Diamond\Box$ and its converse are a consequence of rule $['] \blacktriangleleft \Diamond$ and its converse, using Lemma 2 on $\Diamond\Box\phi \rightsquigarrow \phi \blacktriangleleft \Diamond\Box\phi$ by rule $\rightsquigarrow \blacktriangleleft\Diamond$, and noticing that $v \vDash (\langle x' = \theta\rangle(\neg\chi) \vee \langle x' = \theta\rangle[x' = \theta]\phi)$ is implied by $v \vDash [x' = \theta \,\&\, \chi]\phi$.

$[']\Box\Diamond$ Soundness of $[']\Box\Diamond$ and its converse are a consequence of rule $['] \blacktriangleleft \Box$ and its converse, using Lemma 2 on $\Box\Diamond\phi \rightsquigarrow \phi \blacktriangleleft \Box\Diamond\phi$ by rule $\rightsquigarrow \blacktriangleleft\Box$, and noticing that $v \vDash (\langle x' = \theta\rangle(\neg\chi) \vee \langle x' = \theta\rangle[x' = \theta]\phi)$ is implied by $v \vDash [x' = \theta \,\&\, \chi]\phi$.

$[^*]\Box$ Soundness of $[^*]\Box$ and its converse are a direct consequence of rule $[^*]\sqcap$ and its converse, using Lemma 2 on $\Box\phi \rightsquigarrow \mathsf{true} \sqcap \Box\phi$ by rule $\rightsquigarrow\sqcap$.

$[^*]\Diamond\Box$ Soundness of $[^*]\Diamond\Box$ and its converse are a direct consequence of rule $[^*]\blacktriangleleft$ and its converse, using Lemma 2 on $\Diamond\Box\phi \rightsquigarrow \phi \blacktriangleleft \Diamond\Box\phi$ by rule $\rightsquigarrow \blacktriangleleft\Diamond$.

$[^*]\Box\Diamond$ Soundness of $[^*]\Box\Diamond$ and its converse are a direct consequence of rule $[^*]\blacktriangleleft$ and its converse, using Lemma 2 on $\Box\Diamond\phi \rightsquigarrow \phi \blacktriangleleft \Box\Diamond\phi$ by rule $\rightsquigarrow \blacktriangleleft\Box$.

□

A.2 A somewhat simpler but less efficient system

Another alternative is to suppress all the $[\,]\sqcap$ rules of Fig. 2 (rules $[;]\sqcap$, $[?]\sqcap$, $[:=]\sqcap$, $[']\sqcap$ and $[^*]\sqcap$) and replace them by the $[\,]\Box$ rules of Fig. 3 (rules $[;]\Box$, $[?]\Box$, $[:=]\Box$, $[']\Box$ and $[^*]\Box$) and the following intuitive rule:

$$\frac{[\alpha]\phi \wedge [\alpha]\Box\psi}{[\alpha](\phi \sqcap \Box\psi)}([\,]\sqcap)$$

These changes keep the same proving power. In some way it's nicer because the rules look less complicated, and there is only one more intuitive rule $[\,]\sqcap$, very simple. The system, limited to formulas $[\alpha]\Box\phi$ and $\langle\alpha\rangle\Box\phi$ also reduces to the system presented in dTL [Pla07], [Pla10, chapter 4].

However the duplication of the program α in rule $[\,]\sqcap$ is not desirable. Specifically, an expression of the form $[\alpha;\beta]\Box\phi$ now reduces to $[\alpha]([\beta]\Box\phi\sqcap\Box\phi)$ (using rule $[;]\Box$), which itself reduces to $[\alpha][\beta]\Box\phi \wedge [\alpha]\Box\phi$ (using rule $[\,]\sqcap$). The proof is now split into two very similar subproofs; avoiding this duplication of α in the system dTL^2 presented in Fig. 3 solves an open problem introduced in the dTL work [Pla07], [Pla10, chapter 4].

Proof of the Soundness of rule $[\,]\sqcap$

Proof. To prove soundness of $[\,]\sqcap$ and its converse, we just need to distinguish between terminating and nonterminating traces. Assume $v \vDash [\alpha]\phi \wedge [\alpha]\Box\psi$, and let σ be a trace of $\tau(\alpha)$ starting with v. If σ terminates, then $\sigma \vDash \phi$ and $\sigma \vDash \Box\psi$, therefore $\sigma \vDash \phi \sqcap \Box\psi$. Otherwise, σ does not terminate and $\sigma \vDash \Box\psi$ which is sufficient to prove $\sigma \vDash \phi \sqcap \Box\psi$. Conversely, assume $v \vDash [\alpha](\phi \sqcap \Box\psi)$. To prove $[\alpha]\phi$ we need only consider terminating traces; for any terminating trace σ of $\tau(\alpha)$ starting

$$\pi \sim \pi \text{ (reflexivity)} \qquad \frac{\pi_2 \sim \pi_1}{\pi_1 \sim \pi_2} \text{ (symmetry)} \qquad \frac{\pi_1 \sim \pi_2 \quad \pi_2 \sim \pi_3}{\pi_1 \sim \pi_3} \text{ (transitivity)}$$

$$\neg\Box\pi \sim \Diamond\neg\pi \ (\sim\neg\Box) \qquad \Box\Box\pi \sim \Box\pi \ (\sim\Box\Box) \qquad \Box\Diamond\Box\phi \sim \Diamond\Box\phi \ (\sim\Diamond\Box)$$

$$\neg\Diamond\pi \sim \Box\neg\pi \ (\sim\neg\Diamond) \qquad \Diamond\Diamond\pi \sim \Diamond\pi \ (\sim\Diamond\Diamond) \qquad \Diamond\Box\Diamond\phi \sim \Box\Diamond\phi \ (\sim\Box\Diamond)$$

$$\frac{\pi_1 \sim \pi_2}{\Box\pi_1 \sim \Box\pi_2} \ (\sim\Box) \qquad \frac{\pi_1 \sim \pi_2}{\Diamond\pi_1 \sim \Diamond\pi_2} \ (\sim\Diamond)$$

Figure 4: Temporal equivalence rules for trace formulas

with v, we have $\sigma \vDash \phi \sqcap \Box\psi$ so last $\sigma \vDash \phi$, therefore $v \vDash [\alpha]\pi$. Now let ρ be a (possibly different and possibly nonterminating) trace of $\tau(\alpha)$ starting with v. By hypothesis $\rho \vDash \phi \sqcap \Box\psi$ so $\rho \vDash \Box\psi$, therefore $v \vDash [\alpha]\Box\psi$. □

B Proofs

B.1 Proof of Lemma 1

Proof. In this proof we define an equivalence relation on trace formulas capturing nice properties of trace formulas — where two formulas are equivalent only if they are satisfied on the same traces (Lemma 6). We further show that every trace formula is equivalent to a trace formula with at most two temporal modalities — specifically of the form ϕ, $\Box\phi$, $\Diamond\phi$, $\Box\Diamond\phi$ or $\Diamond\Box\phi$, and that this simplified trace formula can be found in linear time (Lemma 1).

Let us define an equivalence relation $\sim$ on trace formulas as the smallest equivalence relation (reflexive, symmetric and transitive) satisfying the additional 6 axioms and 2 rules of Fig. 4, where, as usual, π, π_1, π_2 and π_3 range over trace formulas, and ϕ ranges over state formulas. Formally, we say that π_1 and π_2 are equivalent, and we write $\pi_1 \sim \pi_2$, if and only if there is a derivation tree proving $\pi_1 \sim \pi_2$ using only axioms and rules of Fig. 4.

A desirable property is that two equivalent trace formulas are satisfied by exactly the same trace. This is formalized by Lemma 6:

Lemma 6 (Soundness of Equivalence)**.** *If $\pi_1 \sim \pi_2$ then for all traces σ, $\sigma \vDash \pi_1$ if and only if $\sigma \vDash \pi_2$.*

Proof. The proof is by induction on the derivation tree of $\pi_1 \sim \pi_2$. The soundness of reflexivity, symmetry and transitivity is trivial. Rules $\sim\neg\Box$ and $\sim\neg\Diamond$ are standard in temporal logic: if it is not the case that every suffix of σ satisfies π, then there must be a suffix of σ not satisfying π, and vice-versa; similarly, if it is not the case that there exists a suffix of σ satisfying π, then all suffixes of σ must satisfy $\neg\pi$. Rules $\sim\Box\Box$ and $\sim\Diamond\Diamond$ have similarly easy soundness proofs, since the set of suffixes of suffixes of any trace σ is the same as the set of suffixes of σ. For $\sim\Box$, if $\pi_1 \sim \pi_2$, then $\sigma \vDash \Box\pi_1$ if and only if every suffix ρ of σ satisfies π_1, which is true if and only if ρ satisfies π_2 by hypothesis. This holds exactly when $\sigma \vDash \Box\pi_2$. The soundness of $\sim\Diamond$ is similar. Finally, for

$\sim \Diamond\Box$, suppose $\sigma \vDash \Box\Diamond\Box\phi$, then $\sigma \vDash \Diamond\Box\phi$ since σ is a suffix of itself. Conversely if $\sigma \vDash \Diamond\Box\phi$, there exists a suffix σ' of σ such that $\sigma' \vDash \Box\phi$. For any suffix ρ of σ, either σ' is a suffix of ρ and $\rho \vDash \Diamond\Box\phi$, or otherwise ρ is a suffix of σ' and $\rho \vDash \Box\phi$, in particular $\rho \vDash \Diamond\Box\phi$. Therefore $\sigma \vDash \Box\Diamond\Box\phi$. The rule $\sim\Box\Diamond$ is dual and has a similar proof. □

Let us now finish the proof of Lemma 1 and prove that every trace formula π is equivalent to a trace formula with at most two temporal modalities, by giving an explicit method of computing π_2 from π_1. First, using rules $\sim\neg\Box$, $\sim\neg\Diamond$, $\sim\Box$ and $\sim\Diamond$, π_1 is equivalent to a trace formula where negation does not appear in front of any temporal modality, but only in the state formula ϕ at the heart of π. Therefore we can assume without loss of generality that π_1 does not contain any negation in front of a temporal modality. If π_1 does not contain any temporal modality, then π_1 is already a state formula and we are done by reflexivity. If π_1 contains only temporal necessities, then $\pi_1 = \Box\Box\ldots\Box\phi$ for some ϕ. Repeated use of rule $\sim\Box\Box$ proves that $\pi_1 \sim \Box\phi$. Similarly, if π_1 contains only temporal possibilities, repeated use of rule $\sim\Diamond\Diamond$ shows that $\pi_1 \sim \Diamond\phi$ for some ϕ. If π_1 contains both temporal necessities and temporal possibilities, and if the temporal modality the furthest right is a temporal necessity, then $\pi_1 \sim \Diamond\Box\phi$ for some ϕ. Indeed using rules $\sim\Box\Box$, $\sim\Box$ and $\sim\Diamond$, we can show that π_1 is equivalent to a trace formula π_3 finishing in $\Diamond\Box\phi$ for some ϕ. If π_3 is exactly $\Diamond\Box\phi$ then we are done, otherwise we can show that π_3 is equivalent to $\Diamond\Box\phi$ using rules $\sim\Diamond\Box$, $\sim\Box\Box$, $\sim\Box$ and $\sim\Diamond$. Symmetrically, if π_1 contains both temporal necessities and temporal possibilities, and if the temporal modality the furthest right is a temporal possibility, then $\pi_1 \sim \Box\Diamond\phi$ for some ϕ. □

B.2 Proof of Theorem 4

Proof. We prove soundness of each rule individually. Soundness of the system then follows by induction on proof trees.

$[\,]\rightsquigarrow$ Soundness of rule $[\,]\rightsquigarrow$ is a corollary of Lemma 2.

$\langle\,\rangle\rightsquigarrow$ Soundness of rule $\langle\,\rangle\rightsquigarrow$ is also a a corollary of Lemma 2.

$[;]\sqcap$ Assume $v \vDash [\alpha]([\beta](\phi \sqcap \Box\psi) \sqcap \Box\psi)$ and let σ be any trace of $\tau(\alpha;\beta)$ starting with v. If σ is a nonterminating trace of $\tau(\alpha)$, then by hypothesis, $\sigma \vDash \Box\psi$. Otherwise there exists a terminating trace $\rho_1 \in \tau(\alpha)$, and a possibly nonterminating trace $\rho_2 \in \tau(\beta)$ such that $\sigma = \rho_1 \circ \rho_2$. By hypothesis $\rho_1 \vDash \Box\psi$ and $\rho_2 \vDash \phi \sqcap \Box\psi$. Therefore $\rho_1 \circ \rho_2 \vDash \Box\psi$, and if ρ_2 terminates then $\rho_1 \circ \rho_2 \vDash \phi$. In all cases $\sigma \vDash \phi \sqcap \Box\psi$ as desired. Conversely, let us assume that $v \vDash [\alpha;\beta](\phi \sqcap \Box\psi)$ and let ρ_1 be any trace of $\tau(\alpha)$ starting with v. If ρ_1 is nonterminating, then $\rho_1 \in \tau(\alpha;\beta)$, therefore $\rho_1 \vDash \Box\psi$. Otherwise let $\rho_2 \in \tau(\beta)$ such that $\rho_1 \circ \rho_2$ is defined, then $\rho_1 \circ \rho_2 \vDash \phi \sqcap \Box\psi$. In particular $\rho_1 \vDash \Box\psi$ and $\rho_2 \vDash \phi \sqcap \Box\psi$. This is for all ρ_2, therefore $\rho_1 \vDash [\beta](\phi \sqcap \Box\psi) \sqcap \Box\psi$. In all cases $\rho_1 \vDash [\beta](\phi \sqcap \Box\psi) \sqcap \Box\psi$ as desired. Since ρ_1 was arbitrary, $v \vDash [\alpha]([\beta](\phi \sqcap \Box\psi) \sqcap \Box\psi)$.

$[;]\sqcup$ The proof is similar to the proof of $[;]\sqcap$. Assume $v \vDash [\alpha]([\beta](\phi \sqcup \Diamond\psi) \sqcup \Diamond\psi)$ and let σ be any trace of $\tau(\alpha;\beta)$ starting with v. If σ is a nonterminating trace of $\tau(\alpha)$, then by hypothesis, $\sigma \vDash \Diamond\psi$. Otherwise there exist a terminating trace $\rho_1 \in \tau(\alpha)$, and a possibly nonterminating trace $\rho_2 \in \tau(\beta)$ such that $\sigma = \rho_1 \circ \rho_2$. By hypothesis $\rho_1 \vDash \Diamond\psi$ or $\rho_2 \vDash \phi \sqcup \Diamond\psi$. Therefore if ρ_2 terminates then $\rho_1 \circ \rho_2 \vDash \phi$ or $\rho_1 \circ \rho_2 \vDash \Diamond\psi$, and otherwise $\rho_1 \circ \rho_2 \vDash \Diamond\psi$. In all three cases $\sigma \vDash \phi \sqcup \Diamond\psi$ as desired. Conversely, let us assume that $v \vDash [\alpha;\beta](\phi \sqcup \Diamond\psi)$ and let ρ_1 be any trace of $\tau(\alpha)$ starting with v. If ρ_1 is nonterminating, then $\rho_1 \in \tau(\alpha;\beta)$, therefore $\rho_1 \vDash \Diamond\psi$. Otherwise let $\rho_2 \in \tau(\beta)$ such that $\rho_1 \circ \rho_2$ exists, then $\rho_1 \circ \rho_2 \vDash \phi \sqcup \Diamond\psi$. In particular $\rho_1 \vDash \Diamond\psi$ or $\rho_2 \vDash \phi \sqcup \Diamond\psi$. This is for all ρ_2, therefore $\rho_1 \vDash [\beta](\phi \sqcup \Diamond\psi) \sqcup \Diamond\psi$. In all cases $\rho_1 \vDash [\beta](\phi \sqcup \Diamond\psi) \sqcup \Diamond\psi$ as desired.

Before continuing, let us prove the following lemma which will be useful for cases involving the $\blacktriangleleft$ construct:

Lemma 7. *For every trace σ and trace formula of the form $\phi \blacktriangleleft \pi$, the following three statements are equivalent:*

(i) $\sigma \vDash \phi \blacktriangleleft \pi$*;*

(ii) $\rho \vDash \phi \blacktriangleleft \pi$ *for every suffix ρ of σ that is different from* $(\hat{\Lambda})$*;*

(iii) $\rho \vDash \phi \blacktriangleleft \pi$ *for some suffix ρ of σ that is different from* $(\hat{\Lambda})$.

Proof. By definition σ cannot be the empty trace, therefore a suffix ρ of σ always exists, therefore $(ii) \Rightarrow (iii)$ is trivial. Let us proceed by distinguishing whether σ is a finite, error or infinite trace; in each case we prove that $(i) \Rightarrow (ii)$ and $(iii) \Rightarrow (i)$.

If σ is finite, then by definition $\sigma \vDash \phi \blacktriangleleft \pi$ if and only if $\mathsf{last}\,\sigma \vDash \phi$. For any suffix ρ of σ, $\mathsf{last}\,\rho = \mathsf{last}\,\sigma$, therefore this is true if and only if $\rho \vDash \phi \blacktriangleleft \pi$, which concludes the case for finite σ.

If σ is an error trace, then any suffix ρ of σ is an error trace. Let σ' be the prefix of σ obtained from σ by removing its last state $\hat{\Lambda}$, and similarly let ρ' be the prefix of ρ obtained from ρ by removing its last state $\hat{\Lambda}$. As we have seen in Remark 1, if π is of the form $\Diamond\Box\phi$ or $\Box\Diamond\phi$, $\sigma \vDash \pi$ if and only if $\mathsf{last}\,\sigma' \vDash \phi$. Since ρ' is a suffix of σ', this is true if and only if $\mathsf{last}\,\rho' \vDash \phi$, which is itself true if and only if $\rho \vDash \pi$. This concludes the case.

Finally, if σ is an infinite trace, any suffix ρ or σ is also infinite. If π is of the form $\Diamond\Box\psi$, assuming (i), there exists a suffix σ' of σ such that $\sigma' \vDash \Box\psi$. Since σ' and ρ are both suffixes of σ, either σ is a suffix of ρ or ρ is a suffix of σ. If σ' is a suffix of ρ then $\rho \vDash \Diamond\Box\psi$, and otherwise ρ is a suffix of σ' therefore $\rho \vDash \Box\psi$, in particular $\rho \vDash \Diamond\Box\psi$. Assuming (iii), if there is a suffix ρ of σ satisfying $\rho \vDash \Diamond\Box\psi$, then there is a suffix σ' of ρ such that $\sigma' \vDash \Box\psi$; this implies $\sigma \vDash \Diamond\Box\psi$ since σ' is also a suffix of σ.

Dually, if π is of the form $\Box\Diamond\psi$, assuming (i), for all suffixes σ' of σ, we have $\sigma' \vDash \Diamond\psi$; this is true in particular for any suffix of any suffix ρ of σ, therefore $\rho \vDash \Diamond\Box\psi$. Conversely, if there is a suffix ρ of σ such that $\rho \vDash \Box\Diamond\psi$, then any suffix of ρ satisfies $\Diamond\psi$. Since any suffix σ' of σ contains a suffix of ρ, we also have $\sigma' \vDash \Diamond\psi$ then $\sigma \vDash \Box\Diamond\psi$. This concludes the proof of the lemma. $\square$

We can now resume the proof of soundness of Theorem 4.

$[;]\blacleft$ Assume $v \vDash [\alpha]([\beta](\phi \blacleft \pi) \blacleft \pi)$ and let σ be any trace of $\tau(\alpha;\beta)$ starting with v. If σ is a nonterminating trace of $\tau(\alpha)$, then by hypothesis, $\sigma \vDash \pi$ so $\sigma \vDash \phi \blacleft \pi$. Otherwise there exists a terminating trace $\rho_1 \in \tau(\alpha)$, and a possibly nonterminating trace $\rho_2 \in \tau(\beta)$ such that $\sigma = \rho_1 \circ \rho_2$. By hypothesis $\rho_2 \vDash \phi \blacleft \pi$ and ρ_2 is a suffix of σ, therefore by Lemma 7, $\sigma \vDash \phi \blacleft \pi$ (since ρ_2 is a trace, it cannot be equal to $(\hat{\Lambda})$). Conversely, let us assume that $v \vDash [\alpha;\beta](\phi \blacleft \pi)$ and let ρ_1 be any trace of $\tau(\alpha)$ starting with v. If ρ_1 is nonterminating, then $\rho_1 \in \tau(\alpha;\beta)$, therefore $\rho_1 \vDash \pi$. Otherwise let $\rho_2 \in \tau(\beta)$ such that $\rho_1 \circ \rho_2$ exists, then $\rho_1 \circ \rho_2 \vDash \phi \blacleft \pi$, which by Lemma 7 implies $\rho_2 \vDash \phi \blacleft \pi$. This is for all ρ_2, therefore $\rho_1 \vDash [\beta](\phi \blacleft \pi) \blacleft \pi$. In all cases $\rho_1 \vDash [\beta](\phi \blacleft \pi) \blacleft \pi$ as desired.

$[\cup]\xi$ For any state v, we have $v \vDash [\alpha]\xi \wedge [\beta]\xi$ if and only if all traces $\sigma \in \tau(\alpha)$ starting with v satisfies $\sigma \vDash \xi$, and all trace $\rho \in \tau(\beta)$ starting with v also satisfies $\rho \vDash \xi$. This is the case if and only if all traces $\sigma \in \tau(\alpha \cup \beta)$ starting with v satisfy $\sigma \vDash \xi$, which in turn is true if and only if $v \vDash [\alpha \cup \beta]\xi$.

$[?]\sqcap$ Assume $v \vDash (\neg\chi \vee \phi) \wedge \psi$ and let σ be any trace of $\tau(?\chi)$ starting with v. If $v \vDash \neg\chi$ then σ is the nonterminating trace $(\hat{v}, \hat{\Lambda})$, and by hypothesis $v \vDash \psi$, therefore $\sigma \vDash \phi \sqcap \Box\psi$. Otherwise σ is the trace $(\hat{v})$, and by hypothesis $v \vDash \phi$ and $v \vDash \psi$, therefore $\sigma \vDash \phi \sqcap \Box\psi$ as well. Conversely, assuming $v \vDash [?\chi](\phi \sqcap \Box\psi)$, then if $v \vDash \neg\chi$, the trace σ is the nonterminating $(\hat{v}, \hat{\Lambda})$ and $v \vDash \psi$. Otherwise $v \vDash \chi$ and σ is the terminating trace $(\hat{v})$, therefore $v \vDash \phi \wedge \psi$. In both cases $v \vDash (\neg\chi \vee \phi) \wedge \psi$ as desired.

$[?]\sqcup$ Assume $v \vDash (\chi \wedge \phi) \vee \psi$ and let σ be any trace of $\tau(?\chi)$ starting with v. If $v \vDash \chi$ and $v \vDash \phi$ then σ is the terminating trace $(\hat{v})$, therefore $\sigma \vDash \phi$ so $\sigma \vDash \phi \sqcup \Diamond\psi$. Otherwise $v \vDash \psi$ and σ is either the trace $\hat{v}$ or the trace $\hat{\Lambda}$; in both cases $\sigma \vDash \Diamond\psi$ so $\sigma \vDash \phi \sqcup \Diamond\psi$. Conversely, assuming $v \vDash [?\chi](\phi \sqcup \Diamond\psi)$, then if $v \vDash \neg\chi$, the trace σ is the nonterminating $(\hat{v}, \hat{\Lambda})$ and $v \vDash \psi$. Otherwise $v \vDash \chi$ and σ is the terminating trace $(\hat{v})$, therefore $v \vDash \phi$ and $v \vDash \psi$. In both cases $v \vDash (\neg\chi \vee \phi) \wedge \psi$ as desired.

$[?]\blacleft\Diamond$ Assume $v \vDash (\chi \wedge \phi) \vee (\neg\chi \wedge \psi)$. If $v \vDash \neg\chi$ then the unique trace of $?\chi$ starting with v is the nonterminating $(\hat{v}, \hat{\Lambda})$, which satisfies $\Diamond\Box\psi$ since $v \vDash \psi$. Otherwise $v \vDash \chi$ and $v \vDash \phi$, and the unique trace of $?\chi$ starting with v is $(\hat{v})$, which is terminating and satisfies $(\hat{v}) \vDash \phi$. In both cases $v \vDash [?\chi](\phi \blacleft \Diamond\Box\psi)$. Conversely, assuming $v \vDash [?\chi](\phi \blacleft \Diamond\Box\psi)$, if $v \vDash \chi$, the unique trace of $?\chi$ starting with v is the terminating $(\hat{v})$, so $(\hat{v}) \vDash \phi$ and $v \vDash \phi$. If $v \vDash \neg\chi$, the unique trace of $?\chi$ starting with v is the nonterminating $(\hat{v}, \hat{\Lambda})$, which satisfies $\Diamond\Box\psi$, therefore $v \vDash \psi$. Therefore in both cases $v \vDash \neg\chi \vee \phi$.

$[?]\blacleft\Box$ Soundness of $[?]\blacleft\Box$ is similar to soundness of $[?]\blacleft\Diamond$.

$[:=]\sqcap$ For any state v, there is a unique, terminating trace of $\tau(x := \theta)$ starting with v, which is $(\hat{v}, \hat{w})$ with $w = v[x \mapsto \mathsf{val}(v, \theta)]$. Therefore $v \vDash [x := \theta](\phi \sqcap \Box\psi)$ if and only if $w \vDash \phi$, $v \vDash \psi$, and $w \vDash \psi$, which is true if and only if $v \vDash \psi \wedge [x := \theta](\phi \wedge \psi)$.

$[:=]\sqcup$ With the same notations as for the $[:=]\sqcap$ case, $v \vDash [x := \theta](\phi \sqcup \Diamond\psi)$ if and only if $w \vDash \phi$, $v \vDash \psi$, or $w \vDash \psi$, which is true if and only if $v \vDash \psi \vee [x := \theta](\phi \vee \psi)$.

$[:=]\blacktriangleleft$ Soundness and its converse are obvious for $[:=]\blacktriangleleft$, as all traces of $\tau(x := \theta)$ are terminating.

$[']\sqcap$ Assume $v \vDash \psi \wedge [x' = \theta \,\&\, \chi](\phi \wedge \psi)$ and let σ be a trace of $\tau(x' = \theta \,\&\, \chi)$ starting with v. If $v \vDash \neg\chi$, then σ is the nonterminating $(\hat{v}, \hat{\Lambda})$, which satisfies $\sigma \vDash \Box\psi$, therefore $\sigma \vDash \phi \sqcap \Box\psi$. Otherwise $\sigma = \{(f)\}$ for a real function f defined on $D = [0, r]$ or $D = [0, +\infty)$ solution of $x' = \theta$ and satisfying χ on its domain of definition. For any restriction f_a of f to $[0, a] \subseteq D$, by definition $\{(f_a)\} \in \tau(x' = \theta \,\&\, \chi)$, therefore by hypothesis any $w_a = v[x \mapsto f(a)]$ satisfies $w_a \vDash \phi \wedge \psi$. All the states of σ are such w_a, therefore $\sigma \vDash \phi \sqcap \Box\psi$. Conversely, assume $v \vDash [x' = \theta \,\&\, \chi](\phi \sqcap \Box\psi)$. By definition there is at least one trace σ of $\tau(x' = \theta \,\&\, \chi)$ starting with v, and it satisfies $\Box\psi$, therefore $v \vDash \psi$. Now proving $[x' = \theta \,\&\, \chi](\phi \wedge \psi)$ only requires us to consider terminating traces. For any terminating $\sigma \in \tau(x' = \theta \,\&\, \chi)$, we have $\sigma \vDash \phi \sqcap \Box\psi$, so in particular $\sigma \vDash \phi \wedge \psi$.

$[']\sqcup$ Assume $v \vDash (\chi \vee \psi) \wedge [x' = \theta \,\&\, (\chi \wedge \neg\psi)]\phi \wedge \langle x' = \theta\rangle(\neg\chi \vee \psi)$ and let σ be a trace of $\tau(x' = \theta \,\&\, \chi)$ starting with v. If $v \vDash \neg\chi$, then σ is the nonterminating $(\hat{v}, \hat{\Lambda})$ with $v \vDash \psi$, therefore $\sigma \vDash \phi \sqcup \Diamond\psi$. Otherwise $\sigma = \{(f)\}$ for a real function f defined on $D = [0, r]$ or $D = [0, +\infty)$ solution of $x' = \theta$ and satisfying χ on D. If σ satisfies $\Diamond\psi$, then $\sigma \vDash \phi \sqcup \Diamond\psi$ and we are done. Otherwise, if σ is terminating, no state of σ ever satisfies ψ, therefore $\sigma \in \tau(x' = \theta \,\&\, (\chi \wedge \neg\psi))$, so by hypothesis $\sigma \vDash \phi$, leading to $\sigma \vDash \phi \sqcup \Diamond\psi$. Finally, in the case where $\sigma \vDash \Diamond\psi$ does not hold, it is not possible to have an infinite σ: such a σ would verify $\chi \wedge \neg\psi$ in all its states, and any trace of $\tau(x' = \theta)$ would be its prefix, contradicting $v \vDash \langle x' = \theta\rangle(\neg\chi \vee \psi)$.

Conversely, let v be a state satisfying $v \vDash [x' = \theta \,\&\, \chi](\phi \sqcup \Diamond\psi)$. First, if $v \vDash \neg\chi$, then the unique trace $\sigma \in \tau(x' = \theta \,\&\, \chi)$ starting with v is $(\hat{v}, \hat{\Lambda})$, and it satisfies $\Diamond\psi$, therefore $v \vDash \psi$. Therefore in all cases $v \vDash \chi \vee \psi$. Second, let us prove $[x' = \theta \,\&\, (\chi \wedge \neg\psi)]\phi$; for this we only need to consider terminating traces. Let σ be a terminating trace of $\tau(x' = \theta \,\&\, (\chi \wedge \neg\psi))$, then in particular $\sigma \in \tau(x' = \theta \,\&\, \chi)$ so $\sigma \vDash \phi \sqcup \Diamond\psi$. Since σ also has $\neg\psi$ as a domain constraint, we cannot have $\sigma \vDash \Diamond\psi$, therefore $\sigma \vDash \phi$. Third and last, we need to prove that either $v \vDash \langle x' = \theta\rangle(\neg\chi)$ or that $v \vDash \langle x' = \theta\rangle\psi$. If $v \vDash \langle x' = \theta\rangle(\neg\chi)$, there is no infinite trace of $\tau(x' = \theta \,\&\, \chi)$ starting with v. Otherwise there exists a unique infinite trace σ of $\tau(x' = \theta \,\&\, \chi)$ starting with v. By hypothesis $\sigma \vDash \Diamond\psi$, therefore ψ has to be true in some state reached by σ, which is the same

as saying that $\langle x' = \theta \rangle \psi$. We have therefore proved that $\langle x' = \theta \rangle (\neg\chi \vee \psi)$, which concludes.

$[']\blacktriangleleft\Diamond$ Assume $v \vDash (\chi \vee \psi) \wedge [x' = \theta \,\&\, \chi]\phi \wedge (\langle x' = \theta \rangle (\neg\chi) \vee \langle x' = \theta \rangle [x' = \theta]\psi)$ and let σ be a trace of $\tau(x' = \theta \,\&\, \chi)$ starting with v. If σ terminates the result is trivial since $v \vDash [x' = \theta \,\&\, \chi]\phi$. If σ is an error trace then $\sigma = (\hat{v}, \hat{\Lambda})$ and $\sigma \vDash \Diamond\Box\psi$, since $v \vDash \psi$ because $v \nvDash \chi$ and $v \vDash \chi \vee \psi$. If σ is an infinite trace, then by hypothesis σ satisfies the domain constraint χ throughout, and since all traces starting with v are prefixes of α, we cannot have $\langle x' = \theta \rangle (\neg\chi)$. Therefore $v \vDash \langle x' = \theta \rangle [x' = \theta]\psi$. Therefore there is a finite trace ρ of $\tau(x' = \theta)$ starting with v such that $\mathsf{last}\,\rho \vDash [x' = \theta]\psi$. By uniqueness [Pla10, appendix B] of the solution of a differential equation, ρ is a prefix of σ; let (i, ζ) be the position of σ such that $\sigma_i(\zeta) = \mathsf{last}\,\rho$, and let σ' be the suffix of σ starting at position (i, ζ). Since $\mathsf{first}\,\sigma' \vDash [x' = \theta]\psi$ and by uniqueness of the solution of the differential equation, $\sigma' \vDash \Box\psi$, therefore $\sigma \vDash \Diamond\Box\psi$. Conversely, assuming $v \vDash [x' = \theta \,\&\, \chi](\phi \blacktriangleleft \Diamond\Box\psi)$, if $v \nvDash \chi$ then σ is the error trace $(\hat{v}, \hat{\Lambda})$, so $\sigma \vDash \Diamond\Box\psi$, therefore $v \vDash \psi$. Therefore in all cases $v \vDash \chi \vee \psi$. Now every terminating trace of $\tau(x' = \theta \,\&\, \chi)$ satisfies ϕ, therefore $v \vDash [x' = \theta \,\&\, \chi]\phi$. If $v \vDash \langle x' = \theta \rangle (\neg\chi)$ then there is no infinite trace in $\tau(x' = \theta \,\&\, \chi)$. Otherwise there is exactly one infinite trace $\sigma \in \tau(x' = \theta \,\&\, \chi)$ and $\sigma \vDash \Diamond\Box\psi$. Therefore there exists a suffix σ' of σ satisfying $\Box\psi$; let (i, ζ) be the position of σ such that $\mathsf{first}\,\sigma' = \sigma_i(\zeta)$. Since σ' is an infinite trace satisfying $\sigma' \vDash \Box\psi$, and by uniqueness of the solution of a differential equation, $\mathsf{first}\,\sigma' \vDash [x' = \theta]\psi$, therefore $v \vDash \langle x' = \theta \rangle [x' = \theta]\psi$.

$[']\blacktriangleleft\Box$ This soundness proof is similar to the soundness of $[']\blacktriangleleft\Diamond$. Assume $v \vDash (\chi \vee \psi) \wedge [x' = \theta \,\&\, \chi]\phi \wedge (\langle x' = \theta \rangle (\neg\chi) \vee [x' = \theta]\langle x' = \theta \rangle \psi)$ and let σ be a trace of $\tau(x' = \theta \,\&\, \chi)$ starting with v. If σ terminates the result is trivial since $v \vDash [x' = \theta \,\&\, \chi]\phi$. If σ is an error trace then $\sigma = (\hat{v}, \hat{\Lambda})$ and $\sigma \vDash \Box\Diamond\psi$, since $v \vDash \psi$ because $v \nvDash \chi$ and $v \vDash \chi \vee \psi$. If σ is an infinite trace, then by hypothesis σ satisfies the domain constraint χ throughout, and since all traces starting with v are prefixes of α, we cannot have $\langle x' = \theta \rangle (\neg\chi)$. Therefore $v \vDash [x' = \theta]\langle x' = \theta \rangle \psi$. Let σ' be any suffix of σ, we need to prove that $\sigma' \vDash \Diamond\psi$. Let (i, ζ) be the position of σ such that $\mathsf{first}\,\sigma' = \sigma_i(\zeta)$, and let ρ be the prefix of σ such that $\mathsf{last}\,\rho = \mathsf{first}\,\sigma'$. Then $\rho \in \tau(x' = \theta)$, therefore $\mathsf{last}\,\rho \vDash \langle x' = \theta \rangle \psi$. Since $\mathsf{last}\,\rho = \mathsf{first}\,\sigma'$ and by uniqueness [Pla10, appendix B] of the solution of a differential equation, this means that $\sigma' \vDash \Diamond\psi$, which entails $\sigma \vDash \Box\Diamond\psi$. Conversely, assuming $v \vDash [x' = \theta \,\&\, \chi](\phi \blacktriangleleft \Box\Diamond\psi)$, if $v \nvDash \chi$ then σ is the error trace $(\hat{v}, \hat{\Lambda})$, so $\sigma \vDash \Box\Diamond\psi$, therefore $v \vDash \psi$. Therefore in all cases $v \vDash \chi \vee \psi$. Now every terminating trace of $\tau(x' = \theta \,\&\, \chi)$ satisfies ϕ, therefore $v \vDash [x' = \theta \,\&\, \chi]\phi$. If $v \vDash \langle x' = \theta \rangle (\neg\chi)$ then there is no infinite trace in $\tau(x' = \theta \,\&\, \chi)$. Otherwise there is exactly one infinite trace $\sigma \in \tau(x' = \theta \,\&\, \chi)$ and $\sigma \vDash \Box\Diamond\psi$. Therefore every suffix σ' of σ satisfies $\Diamond\psi$. Let ρ be a finite trace of $\tau(x' = \theta)$ such that $\mathsf{last}\,\rho \vDash [x' = \theta]\psi$, then by unicity of the solution of a differential equation, ρ is a prefix of σ; let (i, ζ) be the position of σ such that $\mathsf{last}\,\rho = \sigma_i\zeta$, and let σ' be the suffix of σ starting at (i, ζ). Then $\sigma' \vDash \Diamond\psi$ and $\mathsf{first}\,\sigma' = \mathsf{last}\,\rho$, therefore by unicity of the solution of a differential

equation, last $\rho \vDash \langle x' = \theta \rangle \psi$ and $\sigma \vDash [x' = \theta]\langle x' = \theta \rangle \psi$.

$[^*]\sqcap$ Assume $v \vDash \phi \wedge [\alpha^*][\alpha](\phi \sqcap \Box\psi)$ and let σ be a trace of $\tau(\alpha^*)$. If $\sigma \in \tau(\alpha^0) = \tau(?\mathsf{true})$ then $\sigma = (\hat{v})$ and $\sigma \vDash (\phi \sqcap \Box\psi)$ since $v \vDash \phi$ and $v \vDash [\alpha](\phi \sqcap \Box\psi)$, in particular $v \vDash (\phi \wedge \psi)$. Otherwise there exists an $n \geq 1$ such that $\sigma = \sigma_1 \circ \ldots \circ \sigma_n$, where $\sigma_i \in \tau(\alpha)$ for all $i \in \{1, \ldots, n\}$. For all $i \in \{1, \ldots, n\}$, $\sigma_1 \circ \ldots \circ \sigma_{i-1} \in \tau(\alpha^*)$ and $\sigma_i \in \tau(\alpha)$, therefore by hypothesis $\sigma_i \vDash \Box\psi$, therefore $\sigma \vDash \Box\psi$ by gluing. Moreover, if σ terminates then σ_n terminates and by hypothesis $\mathsf{last}\, \sigma_n \vDash \phi$, therefore $\mathsf{last}\, \sigma \vDash \phi$. Therefore $\sigma \vDash \phi \sqcap \Box\psi$. Conversely, if $v \vDash [\alpha^*](\phi \sqcap \Box\psi)$, then in particular $(\hat{v}) \vDash \phi \sqcap \Box\psi$ which implies $v \vDash \phi$. Let σ be a terminating trace of $\tau(\alpha^*)$ starting with v and ρ a trace of $\tau(\alpha)$ starting with $\mathsf{last}\, \sigma$. Then $\sigma \circ \rho \in \tau(\alpha^*)$, therefore by hypothesis $\sigma \circ \rho \vDash \phi \sqcap \Box\psi$, in particular $\rho \vDash \phi \sqcap \Box\psi$.

$[^{*n}]\sqcup$ By definition, the programs α^* and $(?\mathsf{true} \cup \alpha; \alpha^*)$ have the same semantics: $\tau(\alpha^*) = \tau(?\mathsf{true} \cup \alpha^*; \alpha)$. Therefore $v \vDash [\alpha^*](\phi \sqcup \Diamond\psi)$ if and only if $v \vDash [?\mathsf{true} \cup \alpha^*; \alpha](\phi \sqcup \Diamond\psi)$, which by rule $[\cup]\pi$ is true if and only if $v \vDash [?\mathsf{true}](\phi \sqcup \Diamond\psi)$ and $v \vDash [\alpha; \alpha^*](\phi \sqcup \Diamond\psi)$. By rule $[?]\sqcup$, $v \vDash [?\mathsf{true}](\phi \sqcup \Diamond\psi)$ is itself equivalent to $\phi \vee \psi$, therefore $v \vDash [\alpha^*](\phi \sqcup \Diamond\psi)$ if and only if $v \vDash (\phi \vee \psi) \wedge [\alpha; \alpha^*](\phi \sqcup \Diamond\psi)$. But $v \vDash \psi$ implies $v \vDash [\alpha; \alpha^*](\phi \sqcup \Diamond\psi)$, therefore this is equivalent to $v \vDash \psi \vee (\phi \wedge [\alpha; \alpha^*](\phi \sqcup \Diamond\psi))$, which concludes.

$\mathsf{ind}\sqcup$ Assuming $v \vDash \forall^\alpha(\phi \to [\alpha](\phi \sqcup \Diamond\psi))$ and $v \vDash \phi$, let σ be a trace in $\tau(\alpha^*)$. If $\sigma = (\hat{v})$ the result is trivial. Otherwise there exists $n \geq 1$ and $\sigma_1, \ldots, \sigma_n$ such that $\sigma = \sigma_1 \circ \ldots \circ \sigma_n$. If there exists a σ_i such that $\sigma_i \vDash \Diamond\psi$, then $\sigma \vDash \Diamond\psi$ and we are done. Note that this is necessarily the case if σ is nonterminating. Otherwise, for any i, since $\sigma_i \in \tau(\alpha)$, instantiating the premise using its universal closure $\forall^\alpha$, if $\mathsf{first}\, \sigma_i \vDash \phi$ then $\sigma_i \vDash \phi \sqcup \Diamond\psi$; but since $\sigma_i \vDash \Diamond\psi$ does not hold, we have $\mathsf{last}\, \sigma_i \vDash \phi$. Since $v \vDash \phi$, by induction on i, we have $\mathsf{last}\, \sigma \vDash \phi$ which concludes. The universal closure $\forall^\alpha$ is necessary as, otherwise, the premise may behave differently in different states.

$\mathsf{con}\sqcap$ Assume that $v \vDash \forall^\alpha \forall r > 0\, (\varphi(r) \to \langle\alpha\rangle(\varphi(r-1) \sqcap \Box\psi))$ and $v \vDash (\exists r\, \varphi(r)) \wedge \psi$. Then there exists a $d \in \mathbb{R}$ such that $v \vDash \varphi(d)$. Now, the proof is a well-founded induction on d. If $d \leq 0$, we directly have $(\hat{v}) \vDash (\exists r \leq 0\, \varphi(r)) \sqcap \Box\psi$, where $(\hat{v}) \in \tau(\alpha^*)$ for zero repetitions. Otherwise, if $d > 0$, we know that $v \vDash \varphi(d)$ and $v \vDash \varphi(d) \to \langle\alpha\rangle(\varphi(d-1) \sqcap \Box\psi)$, therefore $v \vDash \langle\alpha\rangle(\varphi(d-1) \sqcap \Box\psi)$. Therefore there exists a trace $\sigma_1 \in \tau(\alpha)$ such that $\sigma_1 \vDash \varphi(d-1) \sqcap \Box\psi$. Now $\mathsf{last}\, \sigma_1 \vDash \varphi(d-1)$, therefore if $d - 1 \leq 0$ we are done, otherwise we can construct a similar $\sigma_2 \vDash \varphi(d-2) \sqcap \Box\psi$. By induction we continue until a $d \leq 0$; it is well-founded because d decreases by 1 at each step up to the base case $d \leq 0$. We have thus constructed $\sigma = \sigma_1 \circ \cdots \circ \sigma_n \in \tau(\alpha^*)$, where each $\sigma_i \vDash \Box\psi$, thus $\sigma \vDash \Box\psi$, and $\mathsf{last}\, \sigma_n = \mathsf{last}\, \sigma \vDash (\exists r \leq 0)$. Therefore $\sigma \vDash (\exists r \leq 0\, \varphi(r)) \sqcap \Box\psi$.

$[^*]\blacktriangleleft$ Assume $v \vDash \phi \wedge [\alpha^*][\alpha](\phi \blacktriangleleft \pi)$ and let σ be a trace of $\tau(\alpha^*)$. If $\sigma \in \tau(\alpha^0) = \tau(?\mathsf{true})$ then $\sigma = (\hat{v})$ and $\sigma \vDash \mathsf{last}\, \phi$ since $v \vDash \phi$, so $\sigma \vDash \phi \blacktriangleleft \pi$. Otherwise there exists an $n \geq 1$

such that $\sigma = \sigma_1 \circ \ldots \circ \sigma_n$, where $\sigma_i \in \tau(\alpha)$ for all $i \in \{1, \ldots, n\}$. By hypothesis, $\sigma_1 \circ \ldots \circ \sigma_{n-1} \in \tau(\alpha^*)$, therefore $\sigma_n \vDash \phi \blacktriangleleft \pi$, which using Lemma 7 is enough to prove that $\sigma \vDash \phi \blacktriangleleft \pi$, whether σ terminates or not (since σ_n is a trace, it cannot be equal to $(\hat{\Lambda})$). Conversely, if $v \vDash [\alpha^*](\phi \blacktriangleleft \pi)$, then in particular when executing the loop zero times, $\mathsf{last}\,(\hat{v}) \vDash \phi$ which implies $v \vDash \phi$. Let σ be a terminating trace of $\tau(\alpha^*)$ starting with v and ρ a trace of $\tau(\alpha)$ starting with $\mathsf{last}\,\sigma$. Then $\sigma \circ \rho \in \tau(\alpha^*)$, therefore by hypothesis $\sigma \circ \rho \vDash \phi \blacktriangleleft \pi$, which by Lemma 7 is enough to prove $\rho \vDash \phi \blacktriangleleft \pi$.

□

B.3 Proof of Theorem 5

Proof. The proof relies on the relative completeness of $\mathsf{d}\mathcal{L}$ with respect to FOD [Pla08, Pla12], [Pla10, chapter 2]. Except for programs containing repetitions, the dTL^2 calculus successively reduces temporal properties to nontemporal properties. The temporal rules of dTL^2 transforms normalized trace formulas to simpler normalized trace formulas, i.e., in which the temporal modalities appear after simpler programs, or disappear completely. Additionally, every reduction step is an equivalence, meaning that the premise is equivalent to the conclusion, and Lemma 3 ensures that all trace formulas can be handled by the proof system of dTL^2. Hence, the relative completeness of $\mathsf{d}\mathcal{L}$ directly generalizes to dTL^2 for star-free programs. □

C Dual rules

Sequential Composition

$$\frac{\langle\alpha\rangle(\langle\beta\rangle(\phi \sqcup \Diamond\psi) \sqcup \Diamond\psi)}{\langle\alpha;\beta\rangle(\phi \sqcup \Diamond\psi)}(\langle;\rangle\sqcup) \qquad \frac{\langle\alpha\rangle(\langle\beta\rangle(\phi \sqcap \Box\psi) \sqcap \Box\psi)}{\langle\alpha;\beta\rangle(\phi \sqcap \Box\psi)}(\langle;\rangle\sqcap)$$

$$\frac{\langle\alpha\rangle(\langle\beta\rangle(\phi \blacktriangleleft \pi) \blacktriangleleft \pi)}{\langle\alpha;\beta\rangle(\phi \blacktriangleleft \pi)}(\langle;\rangle\blacktriangleleft)$$

Nondeterministic Choice

$$\frac{\langle\alpha\rangle\xi \vee \langle\beta\rangle\xi}{\langle\alpha \cup \beta\rangle\xi}(\langle\cup\rangle\xi)$$

Test

$$\frac{(\chi \wedge \phi) \vee \psi}{\langle?\chi\rangle(\phi \sqcup \Diamond\psi)}(\langle?\rangle\sqcup) \qquad \frac{(\neg\chi \vee \phi) \wedge (\chi \vee \psi)}{\langle?\chi\rangle(\phi \blacktriangleleft \Box\Diamond\psi)}(\langle?\rangle\blacktriangleleft\Box)$$

$$\frac{(\neg\chi \vee \phi) \wedge \psi}{\langle?\chi\rangle(\phi \sqcap \Box\psi)}(\langle?\rangle\sqcap) \qquad \frac{(\neg\chi \vee \phi) \wedge (\chi \vee \psi)}{\langle?\chi\rangle(\phi \blacktriangleleft \Diamond\Box\psi)}(\langle?\rangle\blacktriangleleft\Diamond)$$

Assignment

$$\frac{\psi \vee \langle x := \theta\rangle(\phi \vee \psi)}{\langle x := \theta\rangle(\phi \sqcup \Diamond\psi)}(\langle:=\rangle\sqcup) \qquad \frac{\psi \wedge \langle x := \theta\rangle(\phi \wedge \psi)}{\langle x := \theta\rangle(\phi \sqcap \Box\psi)}(\langle:=\rangle\sqcap) \qquad \frac{\langle x := \theta\rangle\phi}{\langle x := \theta\rangle(\phi \blacktriangleleft \pi)}(\langle:=\rangle\blacktriangleleft)$$

Ordinary Differential Equation

$$\frac{\psi \vee \langle x' = \theta \,\&\, \chi\rangle(\phi \vee \psi)}{\langle x' = \theta \,\&\, \chi\rangle(\phi \sqcup \Diamond\psi)}(\langle'\rangle\sqcup)$$

$$\frac{(\neg\chi \wedge \psi) \vee \langle x' = \theta \,\&\, (\chi \vee \neg\psi)\rangle\phi \vee [x' = \theta](\chi \wedge \psi)}{\langle x' = \theta \,\&\, \chi\rangle(\phi \sqcap \Box\psi)}(\langle'\rangle\sqcap)$$

$$\frac{(\neg\chi \wedge \psi) \vee \langle x' = \theta \,\&\, \chi\rangle\phi \vee ([x' = \theta]\chi \wedge [x' = \theta]\langle x' = \theta\rangle\psi)}{\langle x' = \theta \,\&\, \chi\rangle(\phi \blacktriangleleft \Box\Diamond\psi)}(\langle'\rangle\blacktriangleleft\Box)$$

$$\frac{(\neg\chi \wedge \psi) \vee \langle x' = \theta \,\&\, \chi\rangle\phi \vee ([x' = \theta]\chi \wedge \langle x' = \theta\rangle[x' = \theta]\psi)}{\langle x' = \theta \,\&\, \chi\rangle(\phi \blacktriangleleft \Diamond\Box\psi)}(\langle'\rangle\blacktriangleleft\Diamond)$$

Repetition

$$\frac{\phi \vee \langle\alpha^*\rangle\langle\alpha\rangle(\phi \sqcup \Diamond\psi)}{\langle\alpha^*\rangle(\phi \sqcup \Diamond\psi)}(\langle^*\rangle\sqcup) \qquad \frac{\psi \wedge (\phi \vee \langle\alpha;\alpha^*\rangle(\phi \sqcap \Box\psi))}{\langle\alpha^*\rangle(\phi \sqcap \Box\psi)}(\langle^{*n}\rangle\sqcap)$$

$$\frac{\phi \vee \langle\alpha^*\rangle\langle\alpha\rangle(\phi \blacktriangleleft \pi)}{\langle\alpha^*\rangle(\phi \blacktriangleleft \pi)}(\langle^*\rangle\blacktriangleleft)$$

Figure 5: Dual rules for the rules of Figure 2

Sequential Composition

$$\frac{\langle\alpha\rangle(\langle\beta\rangle\Diamond\phi \sqcup \Diamond\phi)}{\langle\alpha;\beta\rangle\Diamond\phi}(\langle;\rangle\Diamond) \qquad \frac{\langle\alpha\rangle(\langle\beta\rangle\Box\phi \sqcap \Box\phi)}{\langle\alpha;\beta\rangle\Box\phi}(\langle;\rangle\Box)$$

$$\frac{\langle\alpha\rangle(\langle\beta\rangle\Box\Diamond\phi \blacktriangleleft \Box\Diamond\phi)}{\langle\alpha;\beta\rangle\Box\Diamond\phi}(\langle;\rangle\Box\Diamond) \qquad \frac{\langle\alpha\rangle(\langle\beta\rangle\Diamond\Box\phi \blacktriangleleft \Diamond\Box\phi)}{\langle\alpha;\beta\rangle\Diamond\Box\phi}(\langle;\rangle\Diamond\Box)$$

Nondeterministic Choice

$$\frac{\langle\alpha\rangle\pi \vee \langle\beta\rangle\pi}{\langle\alpha \cup \beta\rangle\pi}(\langle\cup\rangle\pi)$$

Test

$$\frac{\phi}{\langle?\chi\rangle\Diamond\phi}(\langle?\rangle\Diamond) \qquad \frac{\phi}{\langle?\chi\rangle\Box\phi}(\langle?\rangle\Box) \qquad \frac{\phi}{\langle?\chi\rangle\Box\Diamond\phi}(\langle?\rangle\Box\Diamond) \qquad \frac{\phi}{\langle?\chi\rangle\Diamond\Box\phi}(\langle?\rangle\Diamond\Box)$$

Assignment

$$\frac{\phi \vee \langle x := \theta\rangle\phi}{\langle x := \theta\rangle\Diamond\phi}(\langle:=\rangle\Diamond) \qquad \frac{\phi \wedge \langle x := \theta\rangle\phi}{\langle x := \theta\rangle\Box\phi}(\langle:=\rangle\Box)$$

$$\frac{\langle x := \theta\rangle\phi}{\langle x := \theta\rangle\Box\Diamond\phi}(\langle:=\rangle\Box\Diamond) \qquad \frac{\langle x := \theta\rangle\phi}{\langle x := \theta\rangle\Diamond\Box\phi}(\langle:=\rangle\Diamond\Box)$$

Differential Equation

$$\frac{\phi \vee \langle x' = \theta \,\&\, \chi\rangle\phi}{\langle x' = \theta \,\&\, \chi\rangle\Diamond\phi}(\langle'\rangle\Diamond) \qquad \frac{\phi}{\langle x' = \theta \,\&\, \chi\rangle\Box\phi}(\langle'\rangle\Box)$$

$$\frac{(\neg\chi \wedge \phi) \vee \langle x' = \theta \,\&\, \chi\rangle\phi}{\langle x' = \theta \,\&\, \chi\rangle\Box\Diamond\phi}(\langle'\rangle\Box\Diamond) \qquad \frac{(\neg\chi \wedge \phi) \vee \langle x' = \theta \,\&\, \chi\rangle\phi}{\langle x' = \theta \,\&\, \chi\rangle\Diamond\Box\phi}(\langle'\rangle\Diamond\Box)$$

Repetition

$$\frac{\langle\alpha^*\rangle\langle\alpha\rangle\Diamond\phi}{\langle\alpha^*\rangle\Diamond\phi}(\langle^*\rangle\Diamond) \qquad \frac{\phi}{\langle\alpha^*\rangle\Box\phi}(\langle^*\rangle\Box)$$

$$\frac{\phi \vee \langle\alpha^*\rangle\langle\alpha\rangle\Box\Diamond\phi}{\langle\alpha^*\rangle\Box\Diamond\phi}(\langle^*\rangle\Box\Diamond) \qquad \frac{\phi \vee \langle\alpha^*\rangle\langle\alpha\rangle\Diamond\Box\phi}{\langle\alpha^*\rangle\Diamond\Box\phi}(\langle^*\rangle\Diamond\Box)$$

Figure 6: Dual rules for the rules of Figure 3

$$\frac{\langle\alpha\rangle\phi \vee \langle\alpha\rangle\Diamond\psi}{\langle\alpha\rangle(\phi \sqcup \Diamond\psi)}(\langle\,\rangle\sqcup)$$

Figure 7: Dual rule of rule $[\,]\sqcap$

D Proof calculus of dL

The dL proof calculus was described in [Pla08, Pla12], [Pla10, chapter 2]. We include it Figure 8 for reference.

$$(\neg\text{r})\ \frac{\phi \vdash}{\vdash \neg\phi} \qquad (\vee\text{r})\ \frac{\vdash \phi, \psi}{\vdash \phi \vee \psi} \qquad (\wedge\text{r})\ \frac{\vdash \phi \quad \vdash \psi}{\vdash \phi \wedge \psi} \qquad (\to\text{r})\ \frac{\phi \vdash \psi}{\vdash \phi \to \psi} \qquad (ax)\ \frac{}{\phi \vdash \phi}$$

$$(\neg\text{l})\ \frac{\vdash \phi}{\neg\phi \vdash} \qquad (\vee\text{l})\ \frac{\phi \vdash \quad \psi \vdash}{\phi \vee \psi \vdash} \qquad (\wedge\text{l})\ \frac{\phi, \psi \vdash}{\phi \wedge \psi \vdash} \qquad (\to\text{l})\ \frac{\vdash \phi \quad \psi \vdash}{\phi \to \psi \vdash} \qquad (cut)\ \frac{\vdash \phi \quad \phi \vdash}{\vdash}$$

$$(\langle;\rangle)\ \frac{\langle\alpha\rangle\langle\beta\rangle\phi}{\langle\alpha;\beta\rangle\phi} \qquad (\langle^{*n}\rangle)\ \frac{\phi \vee \langle\alpha\rangle\langle\alpha^*\rangle\phi}{\langle\alpha^*\rangle\phi} \qquad (\langle:=\rangle)\ \frac{\phi^{\theta_1 \ldots \theta_n}_{x_1 \ldots x_n}}{\langle x_1 := \theta_1, .., x_n := \theta_n\rangle\phi}$$

$$([;])\ \frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi} \qquad ([^{*n}])\ \frac{\phi \wedge [\alpha][\alpha^*]\phi}{[\alpha^*]\phi} \qquad ([:=])\ \frac{\langle x_1 := \theta_1, .., x_n := \theta_n\rangle\phi}{[x_1 := \theta_1, .., x_n := \theta_n]\phi}$$

$$(\langle\cup\rangle)\ \frac{\langle\alpha\rangle\phi \vee \langle\beta\rangle\phi}{\langle\alpha \cup \beta\rangle\phi} \qquad (\langle?\rangle)\ \frac{\chi \wedge \psi}{\langle?\chi\rangle\psi} \qquad (\langle'\rangle)\ \frac{\exists t{\geq}0\, \big((\forall 0{\leq}\tilde{t}{\leq}t\, \langle\mathcal{S}(\tilde{t})\rangle\chi) \wedge \langle\mathcal{S}(t)\rangle\phi\big)}{\langle x_1' = \theta_1, .., x_n' = \theta_n \,\&\, \chi\rangle\phi}\ [1]$$

$$([\cup])\ \frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi} \qquad ([?])\ \frac{\chi \to \psi}{[?\chi]\psi} \qquad ([']) \frac{\forall t{\geq}0\, \big((\forall 0{\leq}\tilde{t}{\leq}t\, \langle\mathcal{S}(\tilde{t})\rangle\chi) \to \langle\mathcal{S}(t)\rangle\phi\big)}{[x_1' = \theta_1, .., x_n' = \theta_n \,\&\, \chi]\phi}\ [1]$$

$$(\forall\text{r})\ \frac{\vdash \phi(s(X_1, .., X_n))}{\vdash \forall x\, \phi(x)}\ [2] \qquad (\exists\text{r})\ \frac{\vdash \phi(X)}{\vdash \exists x\, \phi(x)}\ [4]$$

$$(\exists\text{l})\ \frac{\phi(s(X_1, .., X_n)) \vdash}{\exists x\, \phi(x) \vdash}\ [2] \qquad (\forall\text{l})\ \frac{\phi(X) \vdash}{\forall x\, \phi(x) \vdash}\ [4]$$

$$(\text{i}\forall)\ \frac{\vdash \mathrm{QE}(\forall X\, (\Phi(X) \vdash \Psi(X)))}{\Phi(s(X_1, .., X_n)) \vdash \Psi(s(X_1, .., X_n))}\ [3] \qquad (\text{i}\exists)\ \frac{\vdash \mathrm{QE}(\exists X\, \bigwedge_i(\Phi_i \vdash \Psi_i))}{\Phi_1 \vdash \Psi_1 \quad \ldots \quad \Phi_n \vdash \Psi_n}\ [5]$$

$$([]gen)\ \frac{\vdash \mathrm{Cl}_\forall\, (\phi \to \psi)}{[\alpha]\phi \vdash [\alpha]\psi} \qquad (\langle\rangle gen)\ \frac{\vdash \mathrm{Cl}_\forall\, (\phi \to \psi)}{\langle\alpha\rangle\phi \vdash \langle\alpha\rangle\psi}$$

$$(ind)\ \frac{\vdash \mathrm{Cl}_\forall\, (\phi \to [\alpha]\phi)}{\phi \vdash [\alpha^*]\phi} \qquad (con)\ \frac{\vdash \mathrm{Cl}_\forall\, \forall v{>}0\, (\varphi(v) \to \langle\alpha\rangle\varphi(v-1))}{\exists v\, \varphi(v) \vdash \langle\alpha^*\rangle\exists v{\leq}0\, \varphi(v)}\ [6]$$

[1] t and $\tilde{t}$ are fresh logical variables and $\langle\mathcal{S}(t)\rangle$ is the jump set $\langle x_1 := y_1(t), .., x_n := y_n(t)\rangle$ with simultaneous solutions $y_1, .., y_n$ of the respective differential equations with constant symbols x_i as symbolic initial values.

[2] s is a new (Skolem) function symbol and $X_1, .., X_n$ are all free logical variables of $\forall x\, \phi(x)$.

[3] X is a new logical variable. Further, QE needs to be defined for the formula in the premise.

[4] X is a new logical variable.

[5] Among all open branches, free logical variable X only occurs in the branches $\Phi_i \vdash \Psi_i$. Further, QE needs to be defined for the formula in the premise, especially, no Skolem dependencies on X can occur.

[6] Logical variable v does not occur in α.

Figure 8: dL Rules