15-424: Foundations of Cyber-Physical Systems

Lecture Notes on Differential & Temporal Proofs

André Platzer

Carnegie Mellon University
Lecture 17

1 Introduction

This lecture continues the study of temporal aspects of cyber-physical systems that Lecture 16 on Differential & Temporal Logics started. The trace semantics of hybrid programs as well as the semantics of differential temporal dynamic logic (dTL) [Pla10], a temporal extension of differential dynamic logic dℒ [Pla08, Pla12], have been discussed in said lecture. That was very useful, because dTL gives us a way of expressing CPS correctness properties that depend on their temporal behavior, with the most prominent one being that a CPS is supposed to be safe always at all times in the future. But that alone is not enough, unless we also find a way to verify such temporal properties of CPS to find out whether they are true, not just specify them to state what we would like to be true. This is what today's lecture focuses on: how to prove temporal properties of cyber-physical systems. Needless to say that the axiomatics of temporal properties of CPS will give us a complementary understanding of their syntactic and semantic representation from Lecture 16 by the general principles of the logical trinity of syntax, semantics, and axiomatics.

This lecture is based on [Pla10, Chapter 4], which extends [Pla07]. Proof rules for more general temporal properties of hybrid systems are handled elsewhere [JP14].

The most important learning goals of this lecture are:

Modeling and Control: We get a more detailed understanding for the difference between temporal and nontemporal properties of CPS models. An understanding for the temporal behavior of CPS also has an immediate impact on their simulation challenges, because a number of differences in how suitable CPS models are for simulation purposes are only reflected in their temporal behavior and corresponding temporal properties.

Computational Thinking: This lecture focuses on rigorous reasoning techniques for the temporal aspects of CPS. It also addresses subtle aspects with identifying specifications and critical properties of CPS. Since *Differential temporal dynamic logic* dTL [Pla10] extends differential dynamic logic, we can continue to use the familiar proof rules for its nontemporal parts, but need to develop new proof rules for its new temporal operators. A secondary goal in this lecture is practicing the logical trinity consisting of the relationship of syntax, semantics, and axiomatics. Temporal properties of CPS cause some new phenomena in proof rules that we have not seen before.

CPS Skills: Another secondary but useful goal of today's lecture is to develop an intuition for the question which parts of a proof (and, by duality, which parts of a system) will be affected by temporal properties. This can be helpful for identifying the relevant dynamical aspects for each part of a CPS and understanding the analytic impact of modeling tradeoffs concerning temporal behaviors of CPS.

2 Temporal Proof Rules

When extending a logic, it is not enough to extend just the syntax (Lecture 16) and semantics (Lecture 16). The third part of the logical trinity, the proof rules, also need to be extended to handle the new concepts, that is the temporal modalities of dTL.

This section shows a sequent calculus for verifying temporal specifications of hybrid systems in differential temporal dynamic logic dTL. With the basic idea being to perform a symbolic decomposition, the calculus transforms hybrid programs successively into simpler logical formulas describing their effects. Statements about the temporal behaviour of a hybrid program are successively reduced to corresponding nontemporal statements about the intermediate states. This lecture shows a proof calculus for differential temporal dynamic logic dTL that inherits the proof rules of dL from previous lectures and adds new proof rules for temporal modalities.

Inherited Nontemporal Rules The dTL calculus is presented in Fig. 1 and inherits the (nontemporal) dL proof rules, i.e., the propositional, first-order, dynamic, and global rules from dL. That is, it includes the propositional and quantifier rules from Lecture 6. The dynamic rules ($\langle;\rangle$–[′]) and global rules ($[]gen$,$\langle\rangle gen$,ind,con) for handling nontemporal dynamic modalities are also inherited directly from Lecture 6.

The only minor additional observation is that the rules $[\cup]$,$\langle\cup\rangle$ for nondeterministic choices can be generalized to apply to formulas of the form $[\alpha \cup \beta]\pi$ where π is an arbitrary trace formula, and not just a state formula as in dL. Thus, π may begin with $\Box$ or $\Diamond$, which is why the rules are repeated in this generalized form as $[\cup]\Box$ and $\langle\cup\rangle\Diamond$ in Fig. 1.

Temporal Rules The new temporal rules in Fig. 1 for the dTL calculus successively transform temporal specifications of hybrid programs into nontemporal dL formulas.

Note 1.

$$([\cup]\Box)\ \frac{[\alpha]\pi \wedge [\beta]\pi}{[\alpha \cup \beta]\pi}\ [1] \qquad (\langle\cup\rangle\Diamond)\ \frac{\langle\alpha\rangle\pi \vee \langle\beta\rangle\pi}{\langle\alpha \cup \beta\rangle\pi}\ 1$$

$$([;]\Box)\ \frac{[\alpha]\Box\phi \wedge [\alpha][\beta]\Box\phi}{[\alpha;\beta]\Box\phi} \qquad (\langle;\rangle\Diamond)\ \frac{\langle\alpha\rangle\Diamond\phi \vee \langle\alpha\rangle\langle\beta\rangle\Diamond\phi}{\langle\alpha;\beta\rangle\Diamond\phi}$$

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

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

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

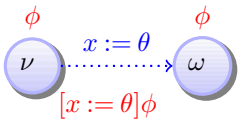
$$([^{*n}]\Box)\ \frac{[\alpha;\alpha^*]\Box\phi}{[\alpha^*]\Box\phi} \qquad (\langle^{*n}\rangle\Diamond)\ \frac{\langle\alpha;\alpha^*\rangle\Diamond\phi}{\langle\alpha^*\rangle\Diamond\phi}$$

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

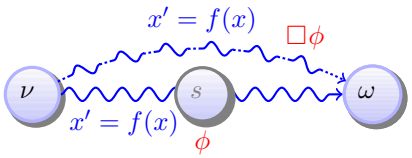
[1] π is a trace formula and—unlike the state formulas ϕ and ψ—may thus begin with a temporal modality $\Box$ or $\Diamond$. That is, π could be of the form ϕ or $\Box\phi$ or $\Diamond\phi$ for a state formula ϕ.

Figure 1: Axiomatization of differential temporal dynamic logic dTL

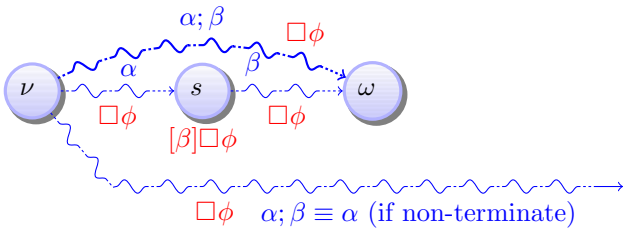
$$\frac{\phi \wedge [x := \theta]\phi}{[x := \theta]\Box\phi}$$



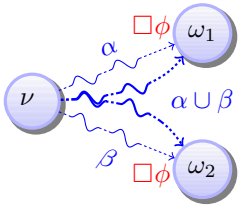
$$\frac{[x' = \theta]\phi}{[x' = \theta]\Box\phi}$$



$$\frac{[\alpha]\Box\phi \wedge [\alpha][\beta]\Box\phi}{[\alpha;\beta]\Box\phi}$$



$$\frac{[\alpha]\Box\phi \wedge [\beta]\Box\phi}{[\alpha \cup \beta]\Box\phi}$$



$$\frac{[\alpha^*][\alpha]\Box\phi}{[\alpha^*]\Box\phi}$$

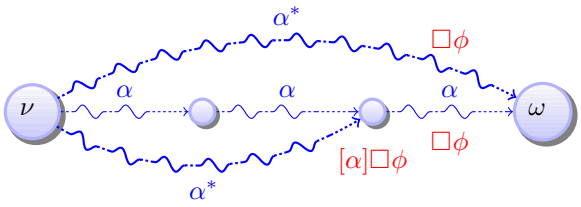


Figure 2: Correspondence of temporal proof rules and trace semantics

The idea underlying this transformation is to decompose hybrid programs and recursively augment intermediate state transitions with appropriate correctness properties that need to be shown for the original temporal property of the whole system. An illustration of the correspondence of a representative set of proof rules for temporal modalities to the trace semantics of hybrid programs (Lecture 16)) is shown in Fig. 2 and will be explained in detail subsequently.

Since a property of a form $[\alpha]\Box\phi$ expresses that formula ϕ is true at all times always throughout every execution of α, we also say informally that ϕ is a (temporal) *invariant* of the hybrid program α. The reason for this name is that ϕ then plays a role somewhat similar to a loop invariant or a differential invariant, except that it holds always throughout the execution of the system.

The most fundamental rule for temporal properties is the one for sequential compositions. Rule [;]□ decomposes invariants of $\alpha;\beta$ (i.e., $[\alpha;\beta]\Box\phi$ holds) into a temporal

invariant that holds throughout α (i.e., $[\alpha]\Box\phi$) and a temporal invariant of β that holds when β is started in *any* final state of α (i.e., $[\alpha]([\beta]\Box\phi)$). The difference of rule [;]□ compared to the dℒ rule [;] thus is that the dTL rule [;]□ also checks the safety invariant ϕ at all (symbolic) states in between the execution of α and β, and recursively in all possible intermediate states because the temporal modality $\Box$ remains both on α (in $[\alpha]\Box\phi$) and on β (in $[\beta]\Box\phi$). See Fig. 2 for an illustration of this proof principle. As an aside, note that the same reasoning principle also works for nonterminating traces of α, because β will just never start in those cases but the subformula $[\alpha]\Box\phi$ already expresses that all nonterminating behaviors of α also satisfy ϕ all the time.

Rule [:=]□ expresses that invariants of assignments just need to hold before and after the discrete change, because there are no other intermediate states. The rule [?]□ is similar, except that tests do not actually lead to any state change, so that ϕ holding before the test is all there is to it, because ϕ will already have to hold after the test $?H$ if it held before. Rule [′]□ can directly reduce invariants of continuous evolutions to nontemporal formulas, because restrictions of solutions of differential equations are themselves solutions of different duration and thus already included in the evolutions of $x' = \theta$. If a property ϕ is true after all solutions of $x' = \theta$ of any duration, then it is also true always throughout every solution of $x' = \theta$ of every duration. In particular, observe that the handling of differential equations within hybrid systems is fully encapsulated within the fragment of dynamic rules from dℒ, which makes differential invariants, differential cuts, differential weakening, and differential ghosts available for temporal properties right away.

The (optional) iteration rule $[*^n]\Box$ can partially unwind loops. It relies on rule [;]□ and is simpler than dℒ rule $[*^n]$, because the other rules will inductively produce a premise to show that ϕ holds in the current state, because of the temporal modality $\Box\phi$. The dual rules $\langle\cup\rangle\Diamond$,$\langle;\rangle\Diamond$,$\langle?\rangle\Diamond$,$\langle:=\rangle\Diamond$,$\langle'\rangle\Diamond$,$\langle*^n\rangle\Diamond$ work similarly.

In dℒ (Lecture 7 on Control Loops & Invariants), the primary means for handling loops are the invariant induction (*ind*) and variant convergence (*con*) rules. Because dTL is built on top of dℒ, the logic dTL takes a different, completely modular approach for verifying temporal properties of loops based on the dℒ capabilities for verifying nontemporal properties of loops. Rules [*]□ and $\langle*\rangle\Diamond$ actually *define* temporal properties of loops inductively. Rule [*]□ expresses that ϕ holds at all times during repetitions of α (i.e., $[\alpha^*]\Box\phi$) iff, *after* repeating α *any* number of times, ϕ holds at all times *during one* execution of α (i.e., $[\alpha^*]([\alpha]\Box\phi)$). See Fig. 2 for an illustration. Dually, $\langle*\rangle\Diamond$ expresses that α holds at some time during repetitions of α (i.e., $\langle\alpha^*\rangle\Diamond\phi$) iff, after some number of repetitions of α, formula ϕ holds at some point during one execution of α (i.e., $\langle\alpha^*\rangle(\langle\alpha\rangle\Diamond\phi)$). In this context, the nontemporal modality $\langle\alpha^*\rangle$ can be thought of as skipping over to the iteration of α during which ϕ actually occurs, as expressed by the nested dTL formula $\langle\alpha\rangle\Diamond\phi$.

Note 2. *The inductive definition rules* $[*]\Box$ *and* $\langle*\rangle\Diamond$ *completely reduce temporal properties of loops to* dTL *properties of standard nontemporal* dL*-modalities such that standard induction (*ind*) or convergence rules (*con*) can be used for the outer nontemporal modality of the loop. Hence, after applying the inductive loop definition rules* $[*]\Box$ *and* $\langle*\rangle\Diamond$*, the standard* dL *loop invariant and variant rules can be used for verifying temporal properties of loops without change, except that the postcondition contains temporal modalities.*

Overall, the temporal repetition rule $[*]\Box$ makes it possible to reduce a temporal property of a repetition to a nontemporal property of a repetition and a temporal property of a non-repetition, both of which are easier. Combining the temporal repetition definition rule $[*]\Box$ with the nontemporal loop induction rule ind' using a loop invariant φ leads to:

$$[*]\Box\frac{ind'\dfrac{\Gamma\vdash\varphi \quad \varphi\vdash[\alpha]\varphi \quad \varphi\vdash[\alpha]\Box\phi}{\Gamma\vdash[\alpha*][\alpha]\Box\phi}}{\Gamma\vdash[\alpha*]\Box\phi}$$

Note that all proof rules for atomic hybrid programs α (i.e., for an α of the form $x:=\theta$, $?H$, or $x'=\theta$) could be summarized into a single rule:

$$\frac{\phi\wedge[\alpha]\phi}{[\alpha]\Box\phi} \qquad \frac{\phi\vee\langle\alpha\rangle\phi}{\langle\alpha\rangle\Diamond\phi} \qquad \text{for atomic } \alpha \tag{1}$$

because the atomic hybrid programs have no temporal behavior that is not already captured at the initial or final states. But the separate proof rules are more efficient, because they leave out parts that are already implied (the postcondition check in the case of $[?]\Box$,$\langle?\rangle\Diamond$ and the precondition check in the case of $[']\Box$,$\langle'\rangle\Diamond$). What happens in the case of differential equations with evolution domain constraints? Does (1) work when α is of the form $x'=\theta\,\&\,H$? Is the following proof rule sound? Or is there a better way?

$$\frac{\phi\wedge[x'=\theta\,\&\,H]\phi}{[x'=\theta\,\&\,H]\Box\phi} \tag{2}$$

Before you read on, see if you can find the answer for yourself.

The proof rule (2) for differential equations with evolution domain constraints looks good but there is a rather subtle issue. Since differential equations are allowed to evolve for zero duration, the first conjunct ϕ seems irrelevant, so that (2) could maybe be simplified to:

$$\frac{[x' = \theta \,\&\, H]\phi}{[x' = \theta \,\&\, H]\Box\phi}$$

It is imperative not to do that! If the system starts in a state ν where the evolution domain constraint H holds ($\nu \models H$), the first conjunct ϕ is, indeed, irrelevant, because $[x' = \theta \,\&\, H]\phi$ implies that all solutions of any duration are safe, including the one of duration zero, which stays at the initial state. However, if the system starts in a state where the evolution domain constraint H does not hold, then there is no solution of $x' = \theta \,\&\, H$ at all, which exempts the premise from having to show anything for that case. Recall that $[x' = \theta \,\&\, \mathit{false}]\phi$ is vacuously true, because ϕ holds after all runs of $x' = \theta \,\&\, \mathit{false}$ just because there are none. Contrast this with the case of $[x' = \theta \,\&\, H]\Box\phi$. If the system starts in a state ν where the evolution domain constraint H does not hold ($\nu \not\models H$), then there is no trace except the error trace $(\hat{\nu}, \hat{\Lambda})$. But the temporal invariant $\Box\phi$ in $[x' = \theta \,\&\, H]\Box\phi$ still requires ϕ to hold at all (non-error) states along that trace, i.e., requires ϕ to hold at ν. No matter what the system, the initial state is always the beginning of every trace, even if the system ends up failing to execute right away. In particular, $[x' = \theta \,\&\, \mathit{false}]\Box\phi$ is equivalent to ϕ.

Consequently, the canonical proof rule for differential equations with evolution domain constraints would be

$$\frac{(H \to [x' = \theta \,\&\, H]\phi) \land (\neg H \to \phi)}{[x' = \theta \,\&\, H]\Box\phi}$$

where the first conjunct considers the case if the evolution domain constraint holds, in which case the temporal postcondition $\Box\phi$ is equivalent to ϕ. The second conjunct considers the case where the differential equation cannot run except into an error state right away, in which case ϕ needs to hold regardless. Now since this first conjunct trivially implies ϕ, it does not change anything if we add an extra ϕ to that. Since the system $x' = \theta \,\&\, H$ has no behavior outside H, it does not change anything if we add $[x' = \theta \,\&\, H]\phi$ in the second conjunct. This leads to the following variation:

$$\frac{(H \to \phi \land [x' = \theta \,\&\, H]\phi) \land (\neg H \to \phi \land [x' = \theta \,\&\, H]\phi)}{[x' = \theta \,\&\, H]\Box\phi}$$

But now the same conditions need to be shown in the premise whether H holds initially or not, which, after simplification, exactly leads to the sound proof rule from (2), which we give the name [&]□:

Note 3. ([&]□) $\dfrac{\phi \land [x' = \theta \,\&\, H]\phi}{[x' = \theta \,\&\, H]\Box\phi}$

Remember that it is imperative to keep the conjunct ϕ in the premise of [&]□.

Even though not necessary, other rules generalize to the temporal case as well such as the generalization rule [] *gen* whose temporal counterpart is gen□.

Note 4 (Temporal generalization)**.**

$$(\text{gen}\Box)\ \frac{\psi \vdash \phi}{[\alpha]\Box\psi \vdash [\alpha]\Box\phi}$$

Similarly for the practical form [] *gen′* of the generalization rule with the temporal counterpart:

$$(\text{gen}\Box)\ \frac{\Gamma \vdash [\alpha]\Box\psi, \Delta \quad \psi \vdash \phi}{\Gamma \vdash [\alpha]\Box\phi, \Delta}$$

Rules for handling $[\alpha]\Diamond\phi$ and $\langle\alpha\rangle\Box\phi$ are briefly discussed in [Pla10] and elaborated in much more detail along with many extensions to more general temporal formulas elsewhere [JP14]. The core challenge is that there is no obvious analogue of the compositional proof rules [;]□,⟨;⟩◇ for the alternation cases $[\alpha]\Diamond\phi$ and $\langle\alpha\rangle\Box\phi$.

3 Temporal Bouncing Ball

Recall the bouncing ball that has served us so well in previous lectures.

$$(2gx \leq 2gH - v^2 \wedge x \geq 0 \wedge g > 0 \wedge H \geq 0 \wedge 1 \geq c \geq 0) \rightarrow [\mathit{ball}^*](0 \leq x \leq H). \tag{3}$$

Use the following abbreviations and the invariant φ from Lecture 11 refining the invariant Lecture 7:

$$\begin{aligned} \mathit{ball} &\equiv x' = v, v' = -g \,\&\, x \geq 0; (?x > 0 \cup (?x = 0; v := -cv)) \\ \Gamma &\equiv g > 0 \wedge H \geq 0 \wedge 1 \geq c \geq 0, \\ \varphi &\equiv 2gx \leq 2gH - v^2 \wedge x \geq 0 \\ \langle x := ..(t)\rangle F &\equiv \langle x := x + vt - \frac{g}{2}t^2; v := v - tg\rangle F. \end{aligned}$$

When simplifying the *ball* dynamics to remove evolution domain constraints:

$$x' = v, v' = -g; (?x > 0 \cup (?x = 0; v := -cv))$$

the proof for the simplified bouncing ball property without evolution domain constraint is shown in Fig. 3. The dℒ proof for the original non temporal bouncing ball property (3) with an evolution domain constraint is shown in Fig. 4. Note that $\langle x := ..(t)\rangle F$ is used as an update in the proof, i.e., proof rules are applied directly to its postcondition F leaving the update around.

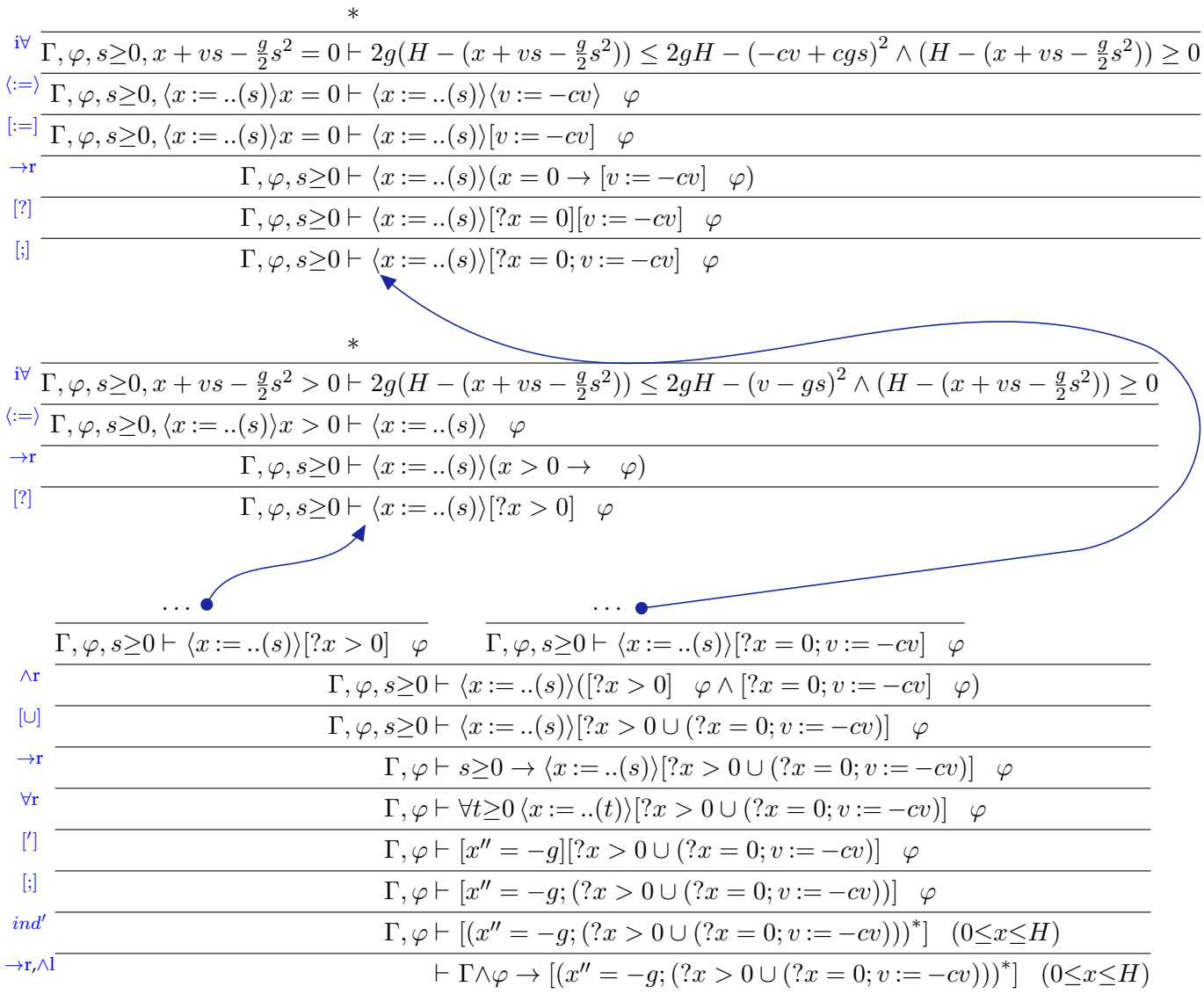


Figure 3: Non-temporal bouncing ball proof (no evolution domain)

$$
\begin{array}{ll}
 & * \\
\text{i}\forall & \Gamma \land \varphi, s{\geq}0, , x + vs - \frac{g}{2}s^2 = 0 \vdash 2g(H - (x + vs - \frac{g}{2}s^2)) \leq 2gH - (-cv + cgs)^2 \land (H - (x + vs - \frac{g}{2}s^2)) \geq 0 \\
\langle{:=}\rangle & \Gamma \land \varphi, s{\geq}0, , \langle x := ..(s)\rangle x = 0 \vdash \langle x := ..(s)\rangle\langle v := -cv\rangle \quad \varphi \\
[:=] & \Gamma \land \varphi, s{\geq}0, , \langle x := ..(s)\rangle x = 0 \vdash \langle x := ..(s)\rangle[v := -cv] \quad \varphi \\
\rightarrow\text{r} & \Gamma \land \varphi, s{\geq}0, \langle x := ..(s)\rangle x \geq 0 \vdash \langle x := ..(s)\rangle(x = 0 \rightarrow [v := -cv] \quad \varphi) \\
[?] & \Gamma \land \varphi, s{\geq}0, \langle x := ..(s)\rangle x \geq 0 \vdash \langle x := ..(s)\rangle[?x = 0][v := -cv] \quad \varphi \\
[;] & \Gamma \land \varphi, s{\geq}0, \langle x := ..(s)\rangle x \geq 0 \vdash \langle x := ..(s)\rangle[?x = 0; v := -cv] \quad \varphi \\
 & * \\
\text{i}\forall & \Gamma \land \varphi, s{\geq}0, , x + vs - \frac{g}{2}s^2 > 0 \vdash 2g(H - (x + vs - \frac{g}{2}s^2)) \leq 2gH - (v - gs)^2 \land (H - (x + vs - \frac{g}{2}s^2)) \geq 0 \\
\langle{:=}\rangle & \Gamma \land \varphi, s{\geq}0, , \langle x := ..(s)\rangle x > 0 \vdash \langle x := ..(s)\rangle \quad \varphi \\
\rightarrow\text{r} & \Gamma \land \varphi, s{\geq}0, \langle x := ..(s)\rangle x \geq 0 \vdash \langle x := ..(s)\rangle(x > 0 \rightarrow \quad \varphi) \\
[?] & \Gamma \land \varphi, s{\geq}0, \langle x := ..(s)\rangle x \geq 0 \vdash \langle x := ..(s)\rangle[?x > 0] \quad \varphi \\
 & \dots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \dots \\
 & \dots \vdash \langle x := ..(s)\rangle[?x > 0] \quad \varphi \qquad \Gamma \land \varphi, s{\geq}0, \langle x := ..(s)\rangle x \geq 0 \vdash \langle x := ..(s)\rangle[?x = 0; v := -cv] \quad \varphi \\
\land\text{r} & \Gamma \land \varphi, s{\geq}0, \langle x := ..(s)\rangle x \geq 0 \vdash \langle x := ..(s)\rangle([?x > 0] \quad \varphi \land [?x = 0; v := -cv] \quad \varphi) \\
[\cup] & \Gamma \land \varphi, s{\geq}0, \langle x := ..(s)\rangle x \geq 0 \vdash \langle x := ..(s)\rangle[?x > 0 \cup (?x = 0; v := -cv)] \quad \varphi \\
\rightarrow\text{r} & \Gamma \land \varphi, s{\geq}0 \vdash \langle x := ..(s)\rangle x \geq 0 \rightarrow \langle x := ..(s)\rangle[?x > 0 \cup (?x = 0; v := -cv)] \quad \varphi \\
\rightarrow\text{r} & \Gamma \land \varphi \vdash s{\geq}0 \rightarrow (\langle x := ..(s)\rangle x \geq 0 \rightarrow \langle x := ..(s)\rangle[?x > 0 \cup (?x = 0; v := -cv)] \quad \varphi) \\
\forall\text{r} & \Gamma \land \varphi \vdash \forall t{\geq}0\, (\langle x := ..(t)\rangle x \geq 0 \rightarrow \langle x := ..(t)\rangle[?x > 0 \cup (?x = 0; v := -cv)] \quad \varphi) \\
['] & \Gamma \land \varphi \vdash [x'' = -g \,\&\, x \geq 0][?x > 0 \cup (?x = 0; v := -cv)] \quad \varphi \\
[;] & \Gamma \land \varphi \vdash [x'' = -g \,\&\, x \geq 0; (?x > 0 \cup (?x = 0; v := -cv))] \quad \varphi \\
ind' & \Gamma \land \varphi \vdash [(x'' = -g \,\&\, x \geq 0; (?x > 0 \cup (?x = 0; v := -cv)))^*] \quad (0{\leq}x{\leq}H) \\
\rightarrow\text{r} & \vdash \Gamma \land \varphi \rightarrow [(x'' = -g \,\&\, x \geq 0; (?x > 0 \cup (?x = 0; v := -cv)))^*] \quad (0{\leq}x{\leq}H)
\end{array}
$$

Figure 4: Nontemporal bouncing ball proof (with evolution domain)

4 Verification Example

Recall the bouncing ball. The proofs from previous lectures or Fig. 4 can be generalized easily to a proof of the temporal property

$$2gx \leq 2gH - v^2 \land x \geq 0 \land g > 0 \land H \geq 0 \land 1 \geq c \geq 0 \\ \rightarrow [(x'' = -g \,\&\, x \geq 0; (?x > 0 \cup (?x = 0; v := -cv)))^*]\Box(0 \leq x \leq H). \quad (4)$$

The only aspect of the proof that changes is that the temporal proof rules in Fig. 1 are used instead of the dynamic proof rules for $\mathsf{d}\mathcal{L}$, and that the resulting extra proof goals for the invariance property at intermediate steps have to be proven.

In contrast, the proof in Fig. 3 for the simplified dynamics without evolution domain restriction $x \geq 0$ cannot be generalized to a proof of the temporal property

$$2gx \leq 2gH - v^2 \land x \geq 0 \land g > 0 \land H \geq 0 \land 1 \geq c \geq 0 \\ \rightarrow [(x'' = -g; (?x > 0 \cup (?x = 0; v := -cv)))^*]\Box(0 \leq x \leq H). \quad (5)$$

because the premise $\Gamma, \varphi \vdash [x'' = -g]\varphi$ resulting from the bottom-most occurrence of a sequential composition rule cannot be shown. This difference in provability is for good reasons. The property in (4) is valid, but the property in (5) is not! While there was no noticeable semantical difference between the nontemporal $\mathsf{d}\mathcal{L}$ counterparts of the properties (4) versus (5), there is a decisive difference between the corresponding temporal properties (5) and (4). Because there is no evolution domain restriction in (5), its hybrid program does not prevent continuous evolution to a negative height under the floor ($x < 0$), for which $0 \leq x \leq H$ does not hold.

The reason for this discrepancy of the temporal version compared to the nontemporal versions thus is that the nontemporal modalities do not "see" the temporary violation of $0 \leq x \leq H$. Such a temporary violation of $0 \leq x$ during the continuous evolution does not produce a successful run of the hybrid program, because it is blocked by the subsequent tests $?x = 0$ and $?x > 0$. A state with negative height fails both tests. While this behaviour does not give a successful program transition of $(\nu, \omega) \in \rho(ball)$ by Lecture 3 so that the proof in Fig. 3 is correct, the behaviour still gives a valid trace $\sigma \in \tau(ball)$ by Lecture 16. This trace σ is a partial trace, because it ends in a failure state Λ, but it is still one of the traces that $[ball]\Box(0 \leq x \leq H)$ quantifies over (quite unlike $[ball](0 \leq x \leq H)$, which only considers final states of successful traces).

The proof of the temporal bouncing ball formula (5) is shown in Fig. 5, which is a direct counterpart of the nontemporal proof in Fig. 4. Some additional premises are elided for space reasons (marked as $\triangleleft$). The bottom-most use of the rule gen$\Box$ has an additional premise $\varphi \rightarrow 0 \leq x \leq H$, which proves easily. The bottom-most use of rule $[;]\Box$ has an additional premise which proves as in Lecture 7 already, just with an extra conjunct coming from $[\&]\Box$:

$$[\&]\Box \frac{\Gamma \land \varphi \vdash \varphi \land [x'' = -g \,\&\, x \geq 0]\varphi}{\Gamma \land \varphi \vdash [x'' = -g \,\&\, x \geq 0]\Box\varphi}$$

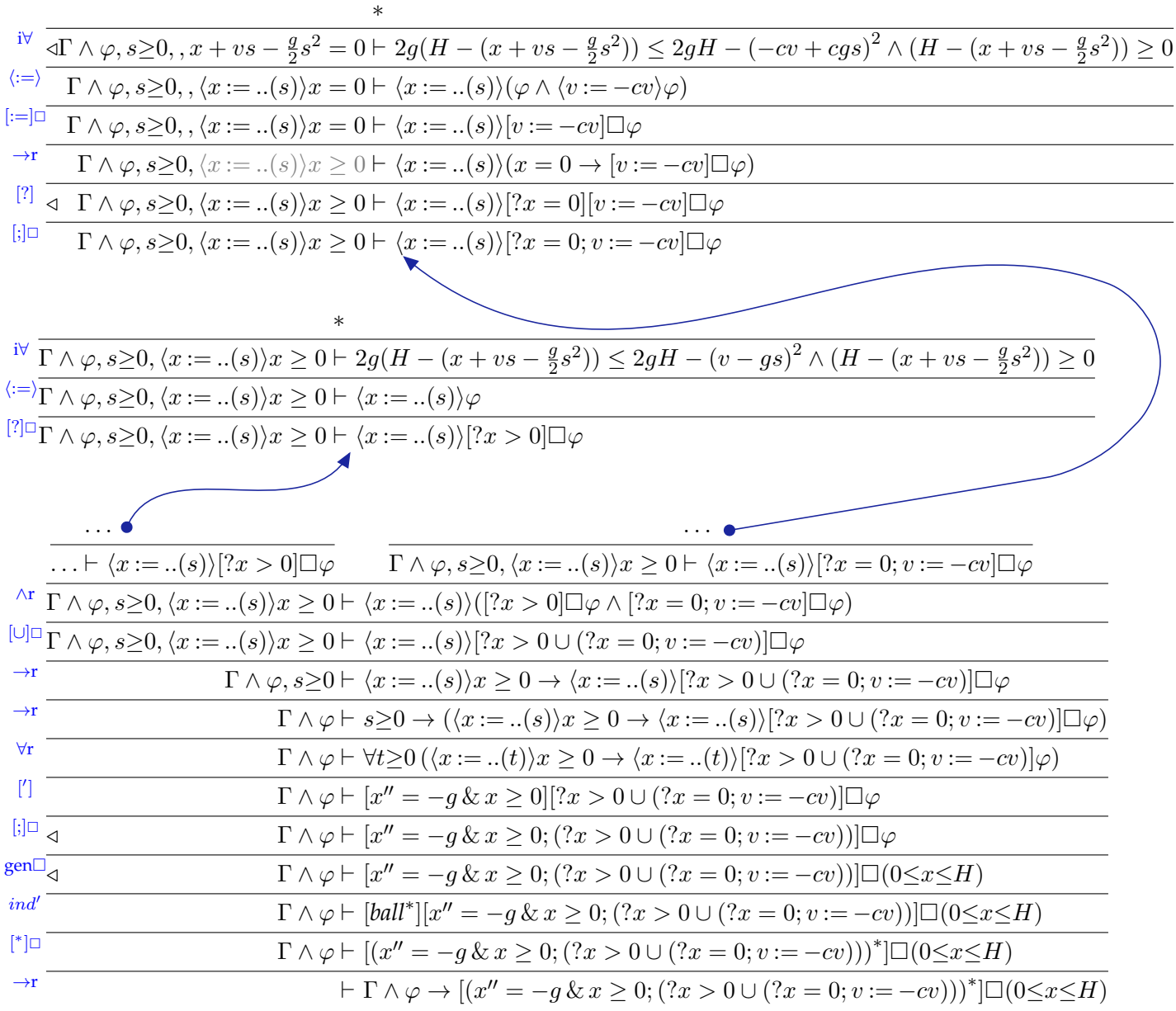


Figure 5: Temporal bouncing ball proof (with evolution domain)

making crucial use of the evolution domain constraint $x \geq 0$. The use of rule [;]□ near the top has an additional premise proving as follows:

$$\text{[?]□}\frac{\Gamma \wedge \varphi, s{\geq}0, \langle x := ..(s)\rangle x \geq 0 \vdash \langle x := ..(s)\rangle \varphi}{\Gamma \wedge \varphi, s{\geq}0, \langle x := ..(s)\rangle x \geq 0 \vdash \langle x := ..(s)\rangle [?x = 0]\Box\varphi}$$

Finally, the top-most use of the $\langle:=\rangle$ rule has a second premise for proving the left conjunct, which is once again:

$$\Gamma \wedge \varphi, s{\geq}0, , \langle x := ..(s)\rangle x = 0 \vdash \langle x := ..(s)\rangle \varphi$$

Finally note that the temporal proof in Fig. 5 was conducted to retain the closest possible resemblance with the nontemporal proof in Fig. 4. But the temporal induction rule [*]□ followed by the nontemporal induction rule ind' does not actually require a proof of the loop invariant φ to hold throughout one iteration of the loop. It would have been sufficient to prove

$$\Gamma \wedge \varphi \vdash [x'' = -g \,\&\, x \geq 0; (?x > 0 \cup (?x = 0; v := -cv))]\Box(0{\leq}x{\leq}H)$$

directly without generalizing the temporal postcondition $\Box(0{\leq}x{\leq}H)$ to $\Box\varphi$ via gen□. That would have led to a proof of the same shape as in Fig. 5, just without gen□ and with easier arithmetic in the end, because $0{\leq}x{\leq}H$ is only a linear constraint on position.

Observe how the temporal analogue of the proof in Fig. 4 shows how every intermediate symbolic state in the hybrid program for the bouncing ball is checked for safety as illustrated by the checkpoint symbol $\wedge$ in the following hybrid program:

$$\wedge\big(\wedge x' = v, v' = -g \,\&\, x \geq 0\wedge; \wedge(\wedge ?x > 0 \cup (\wedge ?x = 0; \wedge v := -cv\wedge))\wedge\big)^*$$

This is another reminder why temporal proofs will check some states repeatedly. There are ways of avoiding such redundancy in the proofs to simplify the computational complexity, but they increase the conceptual complexity of the proof rules, which makes them useful for automation but not necessarily for humans [JP14].

5 Summary

This lecture showed a systematic way of specifying and verifying temporal properties of hybrid systems. The focus was on safety properties that hold always throughout the evolution of the system and are specified as $[\alpha]\Box\phi$ with a mix of a temporal and a dynamic modality instead of just a dynamic modality as in $[\alpha]\phi$. The difference is that $[\alpha]\Box\phi$ includes that safety condition ϕ holds at all intermediate states during all traces of α, whereas $[\alpha]\phi$ only specifies that ϕ holds at the end of each trace of α. This difference matters in systems that have more intermediate states than final states. The difference is insignificant for systems that can "stop anytime", because those will already include all intermediate states of longer system runs as the final state of a corresponding shorter system run. This has been the case in almost all systems studied in this course and is frequently the case in practice.

The systematic way of ensuring safety always throughout the execution of hybrid systems is the use of the dynamic and temporal modality $[\alpha]\Box\phi$, which works whether or not the system has the special structure that allows it to stop anytime. In a nutshell, the temporal proof rules for $[\alpha]\Box\phi$ properties lead to additional branches that correspond to the safety conditions at the respective intermediate state. It can be shown that temporal dTL properties reduce to nontemporal $\mathsf{d}\mathcal{L}$ properties completely [Pla10, Chapter 4], justifying the intimate relation of temporal and nontemporal properties That completeness result made crucial use of the clever form of the $[^*]\Box$ proof rule.

Properties whose temporal counterpart does not hold such as (5) not only indicate that the safety property does not hold throughout the system, but also that the systems might be hard to simulate, because they can run into unsafe dead ends during execution.

Other temporal modalities are more involved and discussed elsewhere [JP14].

Exercises

Exercise 1. Can you give a formula of the following form that is valid?

$$[\alpha]\Box\phi \land \neg[\alpha]\phi$$

Exercise 2. Can you give a temporal box version of the differential invariant proof rule?

Exercise 3. The proof rules in (1) were argued to hold for any atomic hybrid program α. Yet, differential equations with evolution domain constraints were not captured in Fig. 1. Is the case where α is of the form $x' = \theta \,\&\, H$ sound for (1)? Justify why and correct the statement if necessary.

Exercise 4. Augment the nontemporal proof shown in Fig. 4 to a proof of (4). Which of the steps fails when trying to turn Fig. 3 into a proof attempt of (5) and why does that happen? Conduct a proof of (4) that directly proves the postcondition without generalization gen$\Box$.

Exercise 5. Which of the following proof rule attempts are sound? Discuss carefully.

$$\frac{\Gamma \vdash \varphi \quad \varphi \vdash [\alpha]\varphi \quad \varphi \vdash \phi}{\Gamma \vdash [\alpha^*]\Box\phi} \qquad \frac{\Gamma \vdash \varphi \quad \varphi \vdash [\alpha]\varphi \quad \varphi \vdash \Box\phi}{\Gamma \vdash [\alpha^*]\Box\phi} \qquad \frac{\Gamma \vdash \varphi \quad \varphi \vdash [\alpha]\Box\varphi \quad \varphi \vdash \phi}{\Gamma \vdash [\alpha^*]\Box\phi}$$

Exercise 6. In which case does the temporal $[\alpha]\Box\phi$ differ from the nontemporal $[\alpha]\phi$. Hint: consider a number of different forms that α could have.

References

[JP14] Jean-Baptiste Jeannin and André Platzer. dTL^2: Differential temporal dynamic logic with nested temporalities for hybrid systems. In Stéphane Demri, Deepak Kapur, and Christoph Weidenbach, editors, *IJCAR*, volume 8562 of *LNCS*, pages 292–306. Springer, 2014. doi:10.1007/978-3-319-08587-6_22.

[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.