15-424: Foundations of Cyber-Physical Systems

Lecture Notes on Dynamical Systems & Dynamic Axioms

André Platzer

Carnegie Mellon University
Lecture 5

1 Introduction

Lecture 4 demonstrated how useful and crucial CPS contracts are for CPS. Their role and understanding goes beyond dynamic testing, though. In CPS, proven CPS contracts are infinitely more valuable than dynamically tested contracts, because dynamical tests of contracts at runtime of a CPS generally leave open very little flexibility for reacting to them in any safe way. After all, the failure of a contract indicates that some safety condition that was expected to hold is not longer true. Unless provably sufficient safety margin and fallback plans remain, the system is already in trouble then.[1]

Consequently, CPS contracts really shine in relation to how they are proved for CPS. Understanding how to prove CPS contracts requires us to understand the dynamical effects of hybrid programs in more detail. This deeper understanding of the effects of hybrid program statements is not only useful for conducting proofs but also for developing and sharpening our intuition about hybrid programs for CPS. This phenomenon illustrates a more general point that proof and effect (and/or meaning) are intimately linked and that truly understanding effect is ultimately the same as, as well as a prerequisite to, understanding how to prove properties of that effect [Pla12c, Pla12a, Pla10]. You may have seen this point demonstrated amply already in other courses from the Principles of Programming Languages group at CMU.

The route that we choose to get to this level of understanding is one that involves a closer look at dynamical systems and Kripke models, or rather, the effect that hybrid programs have on them. This will enable us to devise authoritative proof principles for differential dynamic logic and hybrid programs [Pla12c, Pla12a, Pla10, Pla08]. While there are many more interesting things to say about dynamical systems and Kripke

[1] Although, in combination with formal verification, the Simplex architecture exploits this relationship of dynamic contracts for safety purposes [SKSC98].

structures, this lecture will limit information to the truly essential parts that are crucial right now and leave more elaboration for later lectures.

More information can be found in [Pla12b, Pla12c] as well as [Pla10, Chapter 2.3].

2 A Proof of Choice (Continued)

Recall the bouncing ball from Lecture 4, with repetition removed just to simplify the discussion for illustration purposes:

$$0 \leq h \wedge h = H \wedge v = 0 \wedge g > 0 \wedge 1 > c \geq 0 \rightarrow \\ \left[h' = v, v' = -g \,\&\, h \geq 0; (?h = 0; v := -cv \cup ?h \neq 0)\right] (0 \leq h \wedge h \leq H) \quad (1)$$

In order to try to prove the above formula, we have convinced ourselves with a number of steps of argumentation that we should try to prove the following two formulas (and many others):

$$0 \leq h \wedge h = H \wedge v = 0 \wedge g > 0 \wedge 1 > c \geq 0 \rightarrow [h' = v, v' = -g \,\&\, h \geq 0](2gh = 2gH - v^2 \wedge g > 0) \\ 2gh = 2gH - v^2 \wedge g > 0 \rightarrow [?h = 0; v := -cv \cup ?h \neq 0] (0 \leq h \wedge h \leq H) \quad (2)$$

In our attempt of proving the latter formula, we used the following principle:

Note 1 (Proving choices). *For a HP that is a nondeterministic choice $\alpha \cup \beta$, we can prove*

$$A \rightarrow [\alpha \cup \beta]B \quad (3)$$

by proving the following dL formulas:

$$A \rightarrow [\alpha]B \qquad \text{and} \qquad A \rightarrow [\beta]B$$

Note 2 (Proving choices: proof-rule style). *Note 1 is captured more concisely in the following proof rule:*

$$(R1)\ \frac{A \rightarrow [\alpha]B \quad A \rightarrow [\beta]B}{A \rightarrow [\alpha \cup \beta]B}$$

If we can prove all premises (above rule bar) of a proof rule, then that proof rule infers the conclusion (below rule bar).

Alas, the way we have been using proof rules so far is the other way around. We had been looking at a formula such as the second formula of (2) that has the shape of the conclusion of a rule such as R1. And then we went on trying to prove the premises of that proof rule instead. This conclusion-to-premise style of using our proof rules is perfectly acceptable and useful as well. Should we ever succeed in proving the premises of R1, that proof rule would allow us to infer its conclusion too. In this way, proof rules are even useful in directing us at which formulas we should try to prove next: the premises of the instantiation of that rule.

Using these thoughts on the second formula of (2), we could prove that formula using proof rule R1 if we would manage to prove both of its premises, which, in this instance, are the following dℒ formulas:

$$\begin{aligned} 2gh = 2gH - v^2 \wedge g > 0 \rightarrow [?h = 0; v := -cv]\,(0 \leq h \wedge h \leq H) \\ 2gh = 2gH - v^2 \wedge g > 0 \rightarrow [?h \neq 0]\,(0 \leq h \wedge h \leq H) \end{aligned} \tag{4}$$

Before proceeding with proofs of (4), revisit the reasoning that led to the principle in Note 2. We said that (3) can be justified by proving that, when assuming A, all runs of α lead to states satisfying B *and* all runs of β lead to B states. Is that argument reflected directly in Note 2?

Kind of, but not quite, because there is a minor difference. Our informal argument assumed A once and concluded both $[\alpha]B$ and $[\beta]B$ from A. The principle captured in Note 2 assumes A to prove $[\alpha]B$ and then, separately, assumes A again to prove $[\beta]B$. These two arguments are clearly closely related, but still slightly different. Can we formalize and follow the original argument directly somehow? Or is Note 2 our only chance?

Following the original argument, we would argue that (3) holds by proving

$$A \rightarrow ([\alpha]B \wedge [\beta]B)$$

or, since the parentheses are superfluous according to the usual precedence rules:

$$A \rightarrow [\alpha]B \wedge [\beta]B \tag{5}$$

Is there a direct way how we can justify going from (3) to (5)? Preferably one that simultaneously justifies going from (3) to the formulas identified in Note 2 as well.

These considerations will take us to a more general and more elegant proof principle than R1, to a more refined understanding of the behavior of nondeterministic choices, and to a way of justifying proof rules as being sound.

3 Dynamic Axioms for Nondeterministic Choices

Recall the semantics of nondeterministic choices from Lecture 3:

$$\rho(\alpha \cup \beta) = \rho(\alpha) \cup \rho(\beta) \tag{6}$$

Remember that $\rho(\alpha)$ is a reachability relation on states, where $(\nu, \omega) \in \rho(\alpha)$ iff HP α can run from state ν to state ω. Let us illustrate graphically what (6) means:

According to $\rho(\alpha)$, a number of states ω_i are reachable by running HP α from some initial state ν. According to $\rho(\beta)$, a number of (possibly other) states ω_i are reachable by running HP β from the same initial state ν. By (6), running $\alpha \cup \beta$ from ν can give us any of those possible outcomes. And there was nothing special about the initial state ν. The same principle holds for all other states.

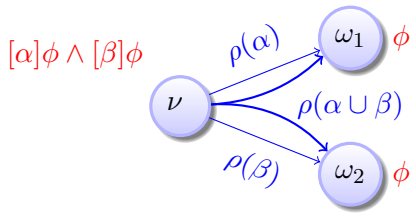


Figure 1: Illustration of transition semantics of $\alpha \cup \beta$

Note 3 ($\cup$). *The nondeterministic choice $\alpha \cup \beta$ can lead to exactly the states to which either α could take us or to which β could take us or to which both could lead. The dynamic effect of a nondeterministic choice $\alpha \cup \beta$ is that running it at any time either results in a behavior of α or of β. So both the behaviors of α and β are possible when running $\alpha \cup \beta$.*

If we want to understand whether and where dL formula $[\alpha \cup \beta]\phi$ is true, we need to understand which states the modality $[\alpha \cup \beta]$ refers to. In which states does ϕ have to be true so that $[\alpha \cup \beta]\phi$ is true in state ν?

By definition of the semantics, ϕ needs to be true in all states that $\alpha \cup \beta$ can reach according to $\rho(\alpha \cup \beta)$ from ν for $[\alpha \cup \beta]\phi$ to be true in ν. Referring to (6) or looking at Fig. 1, shows us that this includes exactly all states that α can reach from ν according to $\rho(\alpha)$, hence $[\alpha]\phi$ has to be true in ν. And that it also includes all states that β can reach from ν, hence $[\beta]\phi$ has to be true in ν.

Consequently,

$$\nu \models [\alpha]\phi \quad \text{and} \quad \nu \models [\beta]\phi \tag{7}$$

are necessary conditions for

$$\nu \models [\alpha \cup \beta]\phi \tag{8}$$

That is, unless (7) holds, (8) cannot possibly hold. So (7) is necessary for (8). Are there any states missing? Are there any states that (8) would require to satisfy ϕ, which (7) does not already ensure to satisfy ϕ? No, because, by (6), $\alpha \cup \beta$ does not admit any behavior that neither α nor β can exhibit. Hence (7) is also sufficient for (8), i.e. (7) implies (8).

Thus, when adopting a more logical language again, this justifies:

$$\nu \models [\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$$

This reasoning did not depend on the particular state ν but holds for all ν. Therefore,

$$\models [\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$$

Exciting! We have just proved our first axiom to be sound:

Lemma 1 ([∪] soundness). *The axiom of choice is* sound, *i.e. all its instances are valid:*

$$([\cup])\ [\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$$

Nondeterministic choices split into their alternatives in axiom [∪]. From right to left: If all α runs lead to states satisfying ϕ (i.e., $[\alpha]\phi$ holds) and all β runs lead to states satisfying ϕ (i.e., $[\beta]\phi$ holds), then all runs of HP $\alpha \cup \beta$, which may choose between following α and following β, also lead to states satisfying ϕ (i.e., $[\alpha \cup \beta]\phi$ holds). The converse implication from left to right holds, because $\alpha \cup \beta$ can run all runs of α and all runs of β, so all runs of α (and of β) lead to states satisfying ϕ if that holds for all runs of $[\beta]\phi$.

From now on, every time we see a formula of the form $[\alpha \cup \beta]\phi$, we can remember that axiom [∪] knows a formula, namely $[\alpha]\phi \wedge [\beta]\phi$ that is equivalent to it. Of course, whenever we find a formula of the form $[\gamma \cup \delta]\psi$, we also remember that axiom [∪] knows a formula, namely $[\gamma]\psi \wedge [\delta]\psi$ that is equivalent to it, just by instantiation of axiom [∪].

Armed with this axiom [∪] at our disposal, we can now easily do a proof step from (3) to (5) just by invoking the equivalence that [∪] justifies. Let's elaborate. We want to prove:

$$A \rightarrow [\alpha \cup \beta]B \tag{3}$$

By [∪], or rather an instance of [∪] formed by using B for ϕ, we know:

$$[\alpha \cup \beta]B \leftrightarrow [\alpha]B \wedge [\beta]B \tag{9}$$

Since (9) is a valid equivalence, replacing the place where the left-hand side of (9) occurs in (3) by the right-hand side of (9) gives us a formula that is equivalent to (3):

$$A \rightarrow [\alpha]B \wedge [\beta]B \tag{5}$$

After all, according to the valid equivalence (9) justified by axiom [∪], (5) can be obtained from (3) just by replacing a formula with one that is equivalent.

Actually, stepping back, the same argument can be made to go from (5) to (3) instead of from (3) to (5). Both ways of using [∪] are perfectly fine. Although the direction that gets rid of the ∪ operator tends to be much more useful, because it made progress (getting rid of an HP operator). Yet axiom [∪] can also be useful in many more situations than rule R1. For example, if want to prove a dℒ formula

$$[\alpha \cup \beta]A \rightarrow B$$

where $[\alpha \cup \beta]$ is on the left-hand side of an implication, then axiom [∪] justifies that it is enough to prove the following dℒ formula instead:

$$[\alpha]A \wedge [\beta]A \rightarrow B$$

This inference cannot be justified with proof rule R1, but would need a separate proof rule such as

$$\text{(R3)}\ \frac{[\alpha]A \wedge [\beta]A \to B}{[\alpha \cup \beta]A \to B}$$

Yet, axiom [∪] justifies both R1 and R3 and many other uses of splitting a boxed choice into a conjunction. Axiom [∪] is, thus, more fundamental.

A general principle behind the dℒ axioms is most noticeable in axiom [∪]. All equivalence axioms of dℒ are primarily intended to be used by reducing the formula on the left to the (structurally simpler) formula on the right. Such a reduction symbolically decomposes a property of a more complicated system into separate properties of easier fragments α and β. This decomposition makes the problem tractable and is good for scalability purposes. For these symbolic structural decompositions, it is very helpful that dℒ is a full logic that is closed under all logical operators, including disjunction and conjunction, for then both sides in [∪] are dℒ formulas again (unlike in Hoare logic [Hoa69]). This also turns out to be an advantage for computing invariants [PC08, PC09, Pla10], which will be discussed much later in this course.

The definition of soundness was not specific to axiom [∪], but applies to all dℒ axioms.

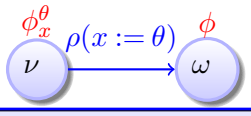
Definition 2 (Soundness)**.** An axiom is *sound* iff all its instances are valid.

4 Dynamic Axioms for Assignments

Axiom [∪] allows us to understand and handle $[\alpha \cup \beta]$ properties. If we find similar axioms for the other operators of hybrid programs, then we have a way of handling "all" other hybrid programs, too.

Consider discrete assignments. Recall from Lecture 4 that:

$$\rho(x := \theta) = \{(\nu, \omega) \ : \ \omega = \nu \text{ except that } [\![x]\!]_\omega = [\![\theta]\!]_\nu\}$$



Lemma 3 ([:=] soundness)**.** *The assignment axiom is sound:*

$$([:=])\ [x := \theta]\phi(x) \leftrightarrow \phi(\theta)$$

Axiom [:=] is Hoare's assignment rule. It uses substitutions to axiomatize discrete assignments. To show that $\phi(x)$ is true after a discrete assignment, axiom [:=] shows that it has been true before, when substituting the affected variable x with its new value θ.

Formula $\phi(\theta)$ is obtained from $\phi(x)$ by *substituting* θ for x at all occurrences of x, provided x does not occur in the scope of a quantifier or modality binding x or a variable of θ.

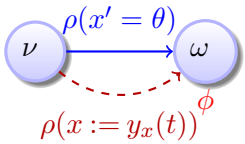
Note 7 (Bound variables). *A modality containing $x := $ or x' outside the scope of tests $?H$ or evolution domain constraints* binds *x, because it may change the value of x. A quantifier $\forall x$ or $\exists x$ also binds variable x.*

Substitutions are defined as usual [Pla10, Chapter 2.5.1].

5 Dynamic Axioms for Differential Equations

Recall from Lecture 4 that

$$\rho(x' = \theta \,\&\, H) = \{(\varphi(0), \varphi(r)) \;:\; \varphi(t) \models x' = \theta \text{ and } \varphi(t) \models H \text{ for all } 0 \leq t \leq r \text{ for a solution } \varphi : [0, r] \to \mathcal{S} \text{ of any duration } r\}$$



One possible approach of proving properties of differential equations is to work with a solution if one is available (and expressible in the logic).

Lemma 4 ([′] soundness). *The solution axiom is sound:*

$$([']) \; [x' = \theta]\phi \leftrightarrow \forall t{\geq}0 \, [x := y(t)]\phi \quad \textit{where } y'(t) = \theta$$

In axiom [′], $y(\cdot)$ is the solution of the symbolic initial-value problem $y'(t) = \theta, y(0) = x$. Solution $y(\cdot)$ is unique since θ is smooth (Lecture 2). Given such a solution $y(\cdot)$, continuous evolution along differential equation $x' = \theta$ can be replaced by a discrete assignment $x := y(t)$ with an additional quantifier for the evolution time t. It goes without saying that variables like t are fresh in [′] and other axioms and proof rules. Notice that conventional initial-value problems are numerical with concrete numbers $x \in \mathbb{R}^n$ as initial values, not symbols x [Wal98]. This would not be enough for our purpose, because we need to consider all states in which the system could start, which may be uncountably many. That is why axiom [′] solves one symbolic initial-value problem, instead, because we could hardly solve uncountable many numerical initial-value problems.

What we have so far about the dynamics of differential equations does not yet help us prove properties of differential equations with evolution domain constraints (a.k.a. continuous programs) $x' = \theta \,\&\, H$. It also does not yet tell us what to do if we cannot solve the differential equation or if the solution is too complicated. We will get to that matter in a much later lecture.

6 Dynamic Axioms for Tests

Recall from Lecture 4 that

$$\rho(?H) = \{(\nu,\nu) \,:\, \nu \models H\}$$

Lemma 5 ([?] soundness)**.** *The test axiom is sound:*

$$([?])\ [?H]\phi \leftrightarrow (H \to \phi)$$

Tests in $[?H]\phi$ are proven by assuming that the test succeeds with an implication in axiom [?], because test $?H$ can only make a transition when condition H actually holds true. In states where test H fails, no transition is possible and the failed attempt to run the system is discarded. If no transition exists, there is nothing to show for $[\alpha]\phi$ formulas, because their semantics requires ϕ to hold in all states reachable by running α, which is vacuously true if no states are reachable. From left to right, axiom [?] for dL formula $[?H]\phi$ assumes that formula H holds true (otherwise there is no transition and thus nothing to show) and shows that ϕ holds after the resulting no-op. The converse implication from right to left is by case distinction. Either H is false, then $?H$ cannot make a transition and there is nothing to show. Or H is true, but then also ϕ is true.

7 Dynamic Axioms for Sequential Compositions

For sequential compositions $\alpha;\beta$, Lecture 4 proposed the use of an intermediate condition E characterizing the intermediate states between α and β by way of the following proof rule:

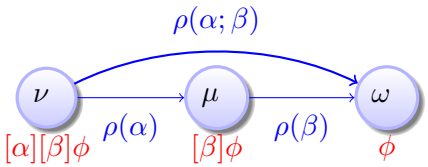
Note 10 (Intermediate conditions as contracts for sequential compositions: proof-rule style)**.** *Intermediate condition contracts for sequential compositions are captured more concisely in the following proof rule:*

$$(R7)\ \frac{A \to [\alpha]E \quad E \to [\beta]B}{A \to [\alpha;\beta]B}$$

~~This proof rule is useful, but it has one blatant annoyance compared to R1 or let alone~~ the simplicity and elegance of [∪]. When using proof rule R7 from the desired conclusion to the premises, it does not say how to choose the intermediate condition E. Using R7 successfully requires us to find the right intermediate condition E, for if we don't, the proof won't succeed as we have seen in Lecture 4. That is a bit much if we have to invent a useful intermediate condition E for every single sequential composition.

Fortunately, there is a much better way that we also identify by investigating the dynamical system resulting from $\alpha;\beta$ and its induced Kripke structure. Recall from Lecture 4 that

$$\rho(\alpha;\beta) = \rho(\beta) \circ \rho(\alpha) = \{(\nu,\omega) : (\nu,\mu) \in \rho(\alpha), (\mu,\omega) \in \rho(\beta)\} \tag{10}$$



By its semantics, the dℒ formula $[\alpha;\beta]\phi$ is true in a state ν iff ϕ is true in all states that $\alpha;\beta$ can reach according to $\rho(\alpha;\beta)$ from ν, i.e. all those states for which $(\nu,\omega) \in \rho(\alpha;\beta)$. Which states are those? And how do they relate to the states reachable by α or by β? They do not relate to those in a way that is as direct as for axiom [∪]. But they still relate, and they do so by way of (10).

Postcondition ϕ has to be true in all states reachable by $\alpha;\beta$ from ν for $[\alpha;\beta]\phi$ to be true at ν. By (10), those are exactly the states ω to which we can get by running β from an intermediate state μ to which we have gotten from ν by running α. Thus, for $[\alpha;\beta]\phi$ to be true at ν it is necessary that ϕ holds in all states ω to which we can get by running β from an intermediate state μ to which we can get by running β from ν. Consequently, $[\alpha;\beta]\phi$ is only true at ν if $[\beta]\phi$ holds in all those intermediate states μ to which we can get from ν by running α. How do we characterize those states? And how can we then express these thoughts in a single logical formula of dℒ?

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

If we want to express that $[\beta]\phi$ holds in all states μ to which we can get to from ν by running α, then that is exactly what truth of dL formula $[\alpha][\beta]\phi$ at ν means, because this is the semantics of the modality $[\beta]$.

Consequently,

$$\nu \models [\alpha][\beta]\phi \rightarrow [\alpha;\beta]\phi$$

Reexamining our argument backwards, we see that the converse implication also holds

$$\nu \models [\alpha;\beta]\phi \rightarrow [\alpha][\beta]\phi$$

The same argument works for all ν, so both implications are even valid.

Lemma 6 ([;] soundness)**.** *The composition axiom is sound:*

$$([;])\ [\alpha;\beta]\phi \leftrightarrow [\alpha][\beta]\phi$$

Proof. Since $\rho(\alpha;\beta) = \rho(\beta) \circ \rho(\alpha)$, we have that $(\nu,\omega) \in \rho(\alpha;\beta)$ iff $(\nu,\mu) \in \rho(\alpha)$ and $(\mu,\omega) \in \rho(\beta)$ for some intermediate state μ. Hence, $\nu \models [\alpha;\beta]\phi$ iff $\mu \models [\beta]\phi$ for all μ with $(\nu,\mu) \in \rho(\alpha)$. That is $\nu \models [\alpha;\beta]\phi$ iff $\nu \models [\alpha][\beta]\phi$. □

Sequential compositions are proven using nested modalities in axiom [;]. From right to left: If, after all α-runs, it is the case that all β-runs lead to states satisfying ϕ (i.e., $[\alpha][\beta]\phi$ holds), then all runs of the sequential composition $\alpha;\beta$ lead to states satisfying ϕ (i.e., $[\alpha;\beta]\phi$ holds). The converse implication uses the fact that if after all α-runs all β-runs lead to ϕ (i.e., $[\alpha][\beta]\phi$), then all runs of $\alpha;\beta$ lead to ϕ (that is, $[\alpha;\beta]\phi$), because the runs of $\alpha;\beta$ are exactly those that first do any α-run, followed by any β-run. Again, it is crucial that dL is a full logic that considers reachability statements as modal operators, which can be nested, for then both sides in [;] are dL formulas.

Axiom [;] directly explains sequential composition $\alpha;\beta$ in terms of a structurally simpler formula, one with nested modal operators but simpler hybrid programs. Again, using axiom [;] by reducing occurrences of its left-hand side to its right-hand side decomposes the formula into structurally simpler pieces, thereby making progress. One of the many ways of using axiom [;] is, therefore, captured in the following proof rule:

$$\text{(R9)}\ \frac{A \rightarrow [\alpha][\beta]B}{A \rightarrow [\alpha;\beta]B}$$

Comparing rule R9 to rule R7, the new rule R9 is much easier to apply, because it does not require us to first provide an intermediate condition E like R7 would. It also does not branch into two premises, which helps keeping the proof lean. Is there a way of reuniting R9 with R7 by using the expressive power of dL?

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

Yes, indeed, there is a very smart choice for the intermediate condition E that makes R7 behave almost as the more efficient R9 would. The clever choice $E \overset{\text{def}}{\equiv} [\beta]B$:

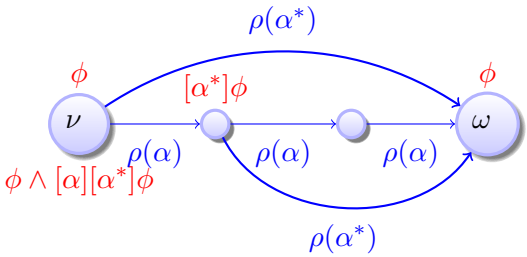
$$\frac{A \to [\alpha][\beta]B \quad [\beta]B \to [\beta]B}{A \to [\alpha;\beta]B}$$

which trivializes the right premise and makes the left premise identical to that of R9.

8 Unwinding Axioms for Loops

Recall from Lecture 4 that

$$\rho(\alpha^*) = \bigcup_{n\in\mathbb{N}} \rho(\alpha^n) \qquad \text{with} \quad \alpha^{n+1} \equiv \alpha^n;\alpha \text{ and } \alpha^0 \equiv ?true$$



Lemma 7 ([*] soundness)**.** *The iteration axiom is sound:*

$$([^*])\ [\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi$$

Axiom [*] is the iteration axiom, which partially unwinds loops. It uses the fact that ϕ always holds after repeating α (i.e., $[\alpha^*]\phi$), if ϕ holds at the beginning (for ϕ holds after zero repetitions then), and if, after one run of α, ϕ holds after every number of repetitions of α, including zero repetitions (i.e., $[\alpha][\alpha^*]\phi$). So axiom [*] expresses that $[\alpha^*]\phi$ holds iff ϕ holds immediately and after one or more repetitions of α. The same axiom [*] can be used to unwind loops $N \in \mathbb{N}$ times, which corresponds to Bounded Model Checking [CBRZ01]. If the formula is not valid, a bug has been found, otherwise N increases. An obvious issue with this simple approach is that we can never stop increasing N if the formula is actually valid, because we can never find a bug then. A later lecture will discuss proof techniques for repetitions based on invariants that are not subject to this issue. In particular, axiom [*] is characteristically different from the other axioms discussed in this lecture. Unlike the other axioms, [*] does not exactly get rid of the formula on the left-hand side. It just puts it in a different syntactic place, which does not sound like much progress.[2]

[2] With a much more subtle and tricky analysis, it is possible to prove that [*] still makes progress [Pla13]. But this is out of scope for our course.

9 A Proof of a Bouncing Ball

Now that we have understood so many axioms and proof rules, let us use them to prove the (single-hop) bouncing ball (1):

$$0 \leq h \wedge h = H \wedge v = 0 \wedge g > 0 \wedge 1 > c \geq 0 \rightarrow \\ \big[h' = v, v' = -g \,\&\, h \geq 0; (?h = 0; v := -cv \cup ?h \neq 0)\big]\,(0 \leq h \wedge h \leq H) \quad (1)$$

Before proceeding, let's modify the hybrid program subtly in tow ways so that there's no more evolution domains, because we have not yet understood how to prove differential equations with evolution domains:

$$0 \leq h \wedge h = H \wedge v = 0 \wedge g > 0 \wedge 1 > c \geq 0 \rightarrow \\ \big[h' = v, v' = -g; (?h = 0; v := -cv \cup ?h \geq 0)\big]\,(0 \leq h \wedge h \leq H) \quad (11)$$

To fit things on the page easily, abbreviate

$$A_{h,v} \overset{\text{def}}{\equiv} 0 \leq h \wedge h = H \wedge v = 0 \wedge g > 0 \wedge 1 > c \geq 0$$
$$B_{h,v} \overset{\text{def}}{\equiv} 0 \leq h \wedge h \leq H$$
$$(h'' = -g) \overset{\text{def}}{\equiv} (h' = v, v' = -g)$$

With these abbreviations, (11) is

$$A_{h,v} \rightarrow [h'' = -g; (?h = 0; v := -cv \cup ?h \geq 0)]B_{h,v}$$

Let there be proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A_{h,v} \rightarrow \forall t{\geq}0\,\big((H - \frac{g}{2}t^2 = 0 \rightarrow B_{H-\frac{g}{2}t^2,-c(-gt)}) \wedge (H - \frac{g}{2}t^2 \geq 0 \rightarrow B_{H-\frac{g}{2}t^2,-gt})\big)}{A_{h,v} \rightarrow \forall t{\geq}0\,[h := H - \frac{g}{2}t^2]\big((h = 0 \rightarrow B_{h,-c(-gt)}) \wedge (h \geq 0 \rightarrow B_{h,-gt})\big)}[:=]}{A_{h,v} \rightarrow \forall t{\geq}0\,[h := H - \frac{g}{2}t^2][v := -gt]\big((h = 0 \rightarrow B_{h,-cv}) \wedge (h \geq 0 \rightarrow B_{h,v})\big)}[:=]}{A_{h,v} \rightarrow \forall t{\geq}0\,[h := H - \frac{g}{2}t^2; v := -gt]\big((h = 0 \rightarrow B_{h,-cv,}) \wedge (h \geq 0 \rightarrow B_{h,v})\big)}[;]}{A_{h,v} \rightarrow [h'' = -g]\big((h = 0 \rightarrow B_{h,-cv,}) \wedge (h \geq 0 \rightarrow B_{h,v})\big)}[']}{A_{h,v} \rightarrow [h'' = -g]\big((h = 0 \rightarrow [v := -cv]B_{h,v}) \wedge (h \geq 0 \rightarrow B_{h,v})\big)}[:=]}{A_{h,v} \rightarrow [h'' = -g]\big([?h = 0][v := -cv]B_{h,v} \wedge [?h \geq 0]B_{h,v}\big)}[?],[?]}{A_{h,v} \rightarrow [h'' = -g]\big([?h = 0; v := -cv]B_{h,v} \wedge [?h \geq 0]B_{h,v}\big)}[;]}{A_{h,v} \rightarrow [h'' = -g][?h = 0; v := -cv \cup ?h \geq 0]B_{h,v}}[\cup]}{A_{h,v} \rightarrow [h'' = -g; (?h = 0; v := -cv \cup ?h \geq 0)]B_{h,v}}[;]$$

Since each of the steps in this proof are justified by using one of the $\mathsf{d}\mathcal{L}$ axioms, the conclusion at the very bottom of this derivation is proved if the premise at the very top can be proved. That premise

$$A_{h,v} \rightarrow \forall t{\geq}0\,\big((H - \frac{g}{2}t^2 = 0 \rightarrow B_{H-\frac{g}{2}t^2,-c(-gt)}) \wedge (H - \frac{g}{2}t^2 \geq 0 \rightarrow B_{H-\frac{g}{2}t^2,-gt})\big)$$

expands out to the following formula of first-order real arithmetic by expanding the abbreviations

$$\begin{aligned}0 \leq h \land h = H \land v = 0 \land g > 0 \land 1 > c \geq 0 \rightarrow \\ \forall t{\geq}0 \big((H - \frac{g}{2}t^2 = 0 \rightarrow 0 \leq H - \frac{g}{2}t^2 \land H - \frac{g}{2}t^2 \leq H) \\ \land (H - \frac{g}{2}t^2 \geq 0 \rightarrow 0 \leq H - \frac{g}{2}t^2 \land H - \frac{g}{2}t^2 \leq H)\big)\end{aligned}$$

In this case, this remaining premise can be easily seen to be valid. The first assumption $H - \frac{g}{2}t^2 = 0 \rightarrow \ldots$ in the middle line directly implies the first conjunct of its right-hand side

$$0 \leq H - \frac{g}{2}t^2 \land H - \frac{g}{2}t^2 \leq H$$

and reduces the second conjunct to $0 \leq H$, which the assumption in the first line assumed ($0 \leq h = H$). Similarly, the first assumption $H - \frac{g}{2}t^2 \geq 0$ of the last line implies the first conjunct of its right-hand side

$$0 \leq H - \frac{g}{2}t^2 \land H - \frac{g}{2}t^2 \leq H$$

and the second conjunct holds by assumption $g > 0$ from the first line and the real arithmetic fact that $t^2 \geq 0$.

How first-order logic and first-order real arithmetic formulas such as this one can be proved in general, however, is an interesting topic for a later lecture. For now, we are happy to report that we have just formally verified our very first CPS. Exciting! We have found a proof of (11).

Okay, admittedly, the CPS we just verified was only a bouncing ball. And all we know about it now is that it won't fall through the cracks in the ground nor jump high up to the moon. But most big steps for mankind start with a small step by someone.

Yet, before we get too carried away, we first need to remember that (11) is just a single-hop bouncing ball. So there's still an argument to be made about what happens if the bouncing ball repeats. And a rather crucial argument too, because bouncing balls let loose in the air tend not to jump any higher without hitting the ground first, which is where the model (11) stops prematurely, because it is missing a repetition. So let's put worrying about loops on the agenda for an upcoming lecture.

Yet, there's one more issue with the proof for the bouncing ball that we derived. It works in a somewhat undisciplined chaotic way, by using $\mathsf{d}\mathcal{L}$ axioms all over the place. This liberal proof style can be useful for manual proofs and creative shortcuts. Albeit, since the $\mathsf{d}\mathcal{L}$ axioms are sound, even such a liberal proof is a proof. But liberal proofs are also somewhat unfocused and non-systematic, which makes them unreasonable for automation purposes and also tends to get people lost if the problems at hand are more complex than the single-hop bouncing ball. That is the reason why we will investigate more focused, more systematic, and more algorithmic proofs next.

10 Summary

The differential dynamic logic axioms that we have seen in this lecture are summarized in Fig. 2. There are further axioms and proof rules of differential dynamic logic that later lectures will examine [Pla12c, Pla12a].

Note 13. *The following axioms of* dL *are sound:*

[:=] $[x := \theta]\phi(x) \leftrightarrow \phi(\theta)$

[?] $[?H]\phi \leftrightarrow (H \rightarrow \phi)$

[′] $[x' = \theta]\phi \leftrightarrow \forall t{\geq}0\, [x := y(t)]\phi$ $\qquad (y'(t) = \theta)$

[∪] $[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$

[;] $[\alpha; \beta]\phi \leftrightarrow [\alpha][\beta]\phi$

[*] $[\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi$

Figure 2: Summary of differential dynamic logic axioms from this lecture

Exercises

Exercise 1. Explain why the subtle transformation from (1) to (11) was okay in this case.

Exercise 2. Identify which of the assumptions of (11) are actually required for the proof of (11). Which formulas could we have dropped from $0 \leq h \wedge h = H \wedge v = 0 \wedge g > 0 \wedge 1 > c \geq 0$ and still be able to prove

$$0 \leq h \wedge h = H \wedge v = 0 \wedge g > 0 \wedge 1 > c \geq 0 \rightarrow [h'' = -g; (?h = 0; v := -cv \cup ?h \geq 0)]0 \leq h \wedge h \leq H$$

Exercise 3. Develop an axiom for differential equations with evolution domains in a style that is similar to [′]. That is, develop an axiom for $[x' = \theta \,\&\, H]\phi$. As in [′], you can assume to have a unique solution for the corresponding symbolic initial-value problem.

Exercise 4. All axioms need to be proved to be sound. These lecture notes only did a proper proof for [;]. Turn the informal arguments for the other axioms into proper soundness proofs using the semantics of dL formulas.

Exercise 5. Would the following be a useful replacement for the [*] axiom?

$$[\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha^*]\phi$$

References

[CBRZ01] Edmund M. Clarke, Armin Biere, Richard Raimi, and Yunshan Zhu. Bounded model checking using satisfiability solving. *Form. Methods Syst. Des.*, 19(1):7–34, 2001.

[DBL12] *Proceedings of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25–28, 2012*. IEEE, 2012.

[Hoa69] Charles Antony Richard Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, 1969.

[PC08] André Platzer and Edmund M. Clarke. Computing differential invariants of hybrid systems as fixedpoints. In Aarti Gupta and Sharad Malik, editors, *CAV*, volume 5123 of *LNCS*, pages 176–189. Springer, 2008. doi:10.1007/978-3-540-70545-1_17.

[PC09] André Platzer and Edmund M. Clarke. Computing differential invariants of hybrid systems as fixedpoints. *Form. Methods Syst. Des.*, 35(1):98–120, 2009. Special issue for selected papers from CAV'08. doi:10.1007/s10703-009-0079-8.

[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.

[Pla12a] André Platzer. The complete proof theory of hybrid systems. In *LICS* [DBL12], pages 541–550. doi:10.1109/LICS.2012.64.

[Pla12b] André Platzer. Dynamic logics of dynamical systems. *CoRR*, abs/1205.4788, 2012. arXiv:1205.4788.

[Pla12c] André Platzer. Logics of dynamical systems. In *LICS* [DBL12], pages 13–24. doi:10.1109/LICS.2012.13.

[Pla13] André Platzer. A complete axiomatization of differential game logic for hybrid games. Technical Report CMU-CS-13-100R, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, January, Revised and extended in July 2013.

[SKSC98] Danbing Seto, Bruce Krogh, Lui Sha, and Alongkrit Chutinan. The Simplex architecture for safe online control system upgrades. In *ACC*, volume 6, pages 3504–3508, 1998.

[Wal98] Wolfgang Walter. *Ordinary Differential Equations*. Springer, 1998.