15-424: Foundations of Cyber-Physical Systems

Lecture Notes on Control Loops & Invariants

André Platzer

Carnegie Mellon University
Lecture 7

1 Introduction

Lecture 3 on Choice & Control demonstrated how important control is in CPS and that control loops are a very important feature for making this control happen. Without loops, CPS controllers are limited to short finite sequences of control actions, which are rarely sufficient. With loops, CPS controllers shine, because they can inspect the current state of the system, take action to control the system, let the physics evolve, and then repeat these steps in a loop over and over again to slowly get the state where the controller wants the system to be. Think of programming a robot to drive on a highway. Would you be able to do that without some means of repetition or iteration? Probably not, because you'll need to write a CPS program that monitors the traffic situation frequently and reacts in response to what the other cars do on the highway.

Hybrid programs' way of exercising repetitive control actions is the repetition operator * that can be applied to any hybrid program α. The resulting hybrid program α^* repeats α any number of times, nondeterministically.

More information can be found in [Pla12b, Pla12a] as well as [Pla10, Chapter 2.5.2,2.5.4].

2 Control Loops

Recall the little acrophobic bouncing ball from Lecture 4 on Safety & Contracts.

$$\begin{aligned}
&\texttt{@requires}(0 \le h \land h = H \land v = 0)\\
&\texttt{@requires}(g > 0 \land 1 \ge c \ge 0)\\
&\texttt{@ensures}(0 \le h \land h \le H)\\
&\big(h' = v, v' = -g \,\&\, h \ge 0;\\
&\ \texttt{if}(h = 0)\, v := -cv\big)^*
\end{aligned} \tag{1}$$

The contracts above have been augmented with the ones that we have identified in Lecture 4 by converting the initial contract specification into a logical formula in differential dynamic logic and then identifying the required assumptions to make it true in all states:

$$0 \leq h \land h = H \land v = 0 \land g > 0 \land 1 \geq c \geq 0 \to \\ \big[\big(h' = v, v' = -g \,\&\, h \geq 0;\ \mathtt{if}(h = 0)\, v := -cv\big)^*\big]\,(0 \leq h \land h \leq H) \quad (2)$$

Because we did not want to be bothered by the presence of the additional `if-then-else` operator, which is not officially part of the minimal set of operators of d$\mathcal{L}$, we simplified (2) to:

$$0 \leq h \land h = H \land v = 0 \land g > 0 \land 1 \geq c \geq 0 \to \\ \big[\big(h' = v, v' = -g \,\&\, h \geq 0;\ (?h = 0; v := -cv \cup ?h \neq 0)\big)^*\big]\,(0 \leq h \land h \leq H) \quad (3)$$

In Lecture 4, we had an informal understanding why (3) is valid (true in all states), but no formal proof, albeit we proved a much simplified version of (3) in which we simply threw away the loop. Ignorance is clearly not a correct way of understanding loops. Let's make up for that now by properly proving (3) in the d$\mathcal{L}$ calculus.

Yet, before going for a proof, let us take a step back and understand the role of loops in more general terms. Their semantics has been explored in Lecture 3 on Choice & Control and more formally in Lecture 5 on Dynamical Systems & Dynamic Axioms.

The little bouncing ball had a loop in which physics and its bounce control alternated. The bouncing ball desperately needs a loop for it wouldn't know ahead of time how often it would bounce. When falling from great heights, it bounces quite a bit. The bouncing ball also has a controller, albeit a rather impoverished one. All it could do is inspect the current height, compare it to the ground floor (at height 0) and, if $h = 0$, flip its velocity vector around after a little damping by factor c. That is not a whole lot of flexibility for control choices, but the bouncing ball was still rather proud to serve such an important role in controlling the bouncing ball's behavior. Indeed, without the control action, the ball would never bounce back from the ground but would keep on falling forever—what a frightful thought for the acrophobic bouncing ball. On second thought, the ball would not fall for very long without its controller, because of the evolution domain $h \geq 0$ for physics $h'' = -g \,\&\, h \geq 0$, which would only allow physics to evolve for time zero if the ball is already at height 0, because gravity would otherwise try to pull it further down, except that $h \geq 0$ won't have it. So, in summary, without the bouncing ball's control statement, it would simply fall and then lie flat on the ground without time being allowed to proceed. That would not sound very reassuring and certainly not as much fun as bouncing back up, so the bouncing ball is really quite proud of its control.

This principle is not specific to the bouncing ball, but, rather, quite common in CPS. The controller performs a crucial task, without which physics would not evolve in the way that we want it to. After all, if physics did already always do what we want it to without any input from our side, we would not need a controller in the first place.

Hence, control is crucial and understanding and analyzing its effect on physics one of the primary responsibilities in CPS.

Before proving (3), we apply one more simplification that we have also done in Lecture 5, just to save space on the page. We boldly drop the evolution domain constraint and make up for it by modifying the condition in the second test (Exercise 1):

$$\begin{aligned}&0 \leq h \land h = H \land v = 0 \land g > 0 \land 1 \geq c \geq 0 \to\\ &\qquad\big[\big(h' = v, v' = -g;\ (?h = 0; v := -cv \cup ?h \geq 0)\big)^*\big]\,(0 \leq h \land h \leq H)\end{aligned} \tag{4}$$

Hold on, why is that okay? Doesn't our previous investigation say that the ball could suddenly fall through the cracks in the floor if physics insists on evolving for hours before giving the poor bouncing ball controller a chance to react? To make sure the bouncing ball does not panic in light of this threat, solve Exercise 1 to investigate this.

3 Proofs of Loops

There is a loop in (4). As we have seen, its behavior is crucial to the bouncing ball. So let's prove to understand what it does and to see whether we have to be just as nervous as the bouncing ball about losing it to the earth (if postcondition $0 \leq h$ is not ensured) or to the sky (if $h \leq H$ is not ensured).

Abbreviations have served us well in trying to keep proofs onto one page.

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

With these abbreviations, the bouncing ball formula (4) turns into:

$$A_{h,v} \to [(h'' = -g; (?h = 0; v := -cv \cup ?h \geq 0))^*]B_{h,v} \tag{4}$$

This formula is swiftly turned into the sequent at the top using proof rule →r:

$$\to\!r\,\frac{A_{h,v} \vdash [(h'' = -g; (?h = 0; v := -cv \cup ?h \geq 0))^*]B_{h,v}}{\vdash A_{h,v} \to [(h'' = -g; (?h = 0; v := -cv \cup ?h \geq 0))^*]B_{h,v}}$$

This leaves a loop to be worried about. Inspecting our d$\mathcal{L}$ proof rules from Lecture 6 on Truth there is exactly one that addresses loops:

$$([^{*n}])\ \frac{\phi \land [\alpha][\alpha^*]\phi}{[\alpha^*]\phi}$$

Using this one to continue the sequent derivation proceeds as follows:

$$\text{[}{*^n}\text{]r}\;\dfrac{\text{∧r}\;\dfrac{\dfrac{*}{A_{h,v} \vdash B_{h,v}} \qquad \text{[;]r}\;\dfrac{A_{h,v} \vdash [h''=-g][?h=0;v:=-cv \cup ?h \geq 0][(h''=-g;(?h=0;v:=-cv \cup ?h \geq 0))^*]B_{h,v}}{A_{h,v} \vdash [h''=-g;(?h=0;v:=-cv \cup ?h \geq 0)][(h''=-g;(?h=0;v:=-cv \cup ?h \geq 0))^*]B_{h,v}}}{A_{h,v} \vdash B_{h,v} \wedge [h''=-g;(?h=0;v:=-cv \cup ?h \geq 0)][(h''=-g;(?h=0;v:=-cv \cup ?h \geq 0))^*]B_{h,v}}}{A_{h,v} \vdash [(h''=-g;(?h=0;v:=-cv \cup ?h \geq 0))^*]B_{h,v}}$$

The left subgoal that results from using ∧r closes by very simple arithmetic. The right subgoal is more of a challenge to prove. We can solve the differential equation and proceed using [′]r, which will produce a quantifier that ∀r can handle and leaves us with a sequent that we need to consider further to prove.

4 Loops of Proofs

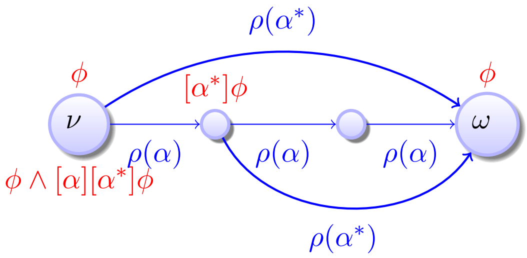
After a lot of proof effort, the above sequent prove continues so that the modalities

$$\dots [h''=-g][?h=0;v:=-cv \cup ?h \geq 0]\psi$$

can be handled. But there is still a loop in the postcondition ψ. How can we prove that postcondition, then? Investigating our proof rules, there is exactly one that addresses loops: $[*^n]$r again. If we use $[*^n]$r again, what will happen?

Recall from Lecture 5

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



Lemma 1 ([*] soundness)**.** *The iteration axiom is sound:*

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

Using proof rule $[*^n]$r on the succedent of a sequent has the same effect as using axiom [*] from left-hand side to right-hand side. Axiom [*] can be used to turn a formula

$$A \rightarrow [\alpha^*]B \tag{5}$$

into

$$A \rightarrow B \wedge [\alpha][\alpha^*]B$$

What happens if we use that axiom [*] again?

Recall that, unlike sequent proof rules such as $[*^n]$r, axioms do not say where they can be used, so we might as well use them anywhere in the middle of the formula. Hence using axiom [*] on the inner loop yields:

$$A \to B \land [\alpha](B \land [\alpha][\alpha^*]B)$$

Let's do that again and use [*] to obtain

$$A \to B \land [\alpha](B \land [\alpha](B \land [\alpha][\alpha^*]B)) \tag{6}$$

This is all very interesting but won't exactly get us any closer to a proof, because we could keep expanding the * star forever that way. How do we ever break out of this loop of never-ending proofs?

Before we get too disillusioned about our progress with [*] so far, notice that (6) still allows us to learn something about α and whether it always satisfies B when repeating α. Since [*] is an equivalence axiom, formula (6) still expresses the same thing as (5), i.e. that B always holds after repeating α when A was true in the beginning. Yet, (6) explicitly singles out the first 3 runs of α. Let's make this more apparent by recalling

$$([]\land)\ [\alpha](B \land \psi) \leftrightarrow [\alpha]B \land [\alpha]\psi$$

Using this valid equivalence turns (6) into

$$A \to B \land [\alpha]B \land [\alpha][\alpha](B \land [\alpha][\alpha^*]B)$$

Using $[]\land$ again gives us

$$A \to B \land [\alpha]B \land [\alpha]([\alpha]B \land [\alpha][\alpha][\alpha^*]B)$$

Using $[]\land$ once more gives

$$A \to B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha][\alpha^*]B \tag{7}$$

Looking at it this way, (7) could be more useful than the original (5), because, even though both are equivalent, (7) explicitly singles out the fact that B has to hold initially, after doing α once, after doing α twice, and that $[\alpha^*]B$ has to hold after doing α three times. Even if we are not quite sure what to make of the latter $[\alpha][\alpha][\alpha][\alpha^*]B$, because it still involves a loop, we are quite certain how to understand and handle the first three:

$$A \to B \land [\alpha]B \land [\alpha][\alpha]B \tag{8}$$

If this formula is not valid, then, certainly, neither is (7) and, thus, neither is the original (5). Hence, if we find a counterexample to (8), we disproved (7) and (5). That can actually be rather useful.

Yet, if (8) is still valid, we do not know whether (7) and (5) are, since they involve stronger requirements (B holds after any number of repetitions of α). What can we do then? Simply unroll the loop once more by using [*] on (6) to obtain

$$A \to B \land [\alpha](B \land [\alpha](B \land [\alpha](B \land [\alpha][\alpha^*]B))) \tag{9}$$

Or, equivalently, use axiom [*] on (7) to obtain the equivalent

$$A \to B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha](B \land [\alpha][\alpha^*]B) \tag{10}$$

By sufficiently many uses of axiom []∧, (9) and (10) are both equivalent to

$$A \to B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha]B \land [\alpha][\alpha][\alpha^*]B \tag{11}$$

which we can again examine to see if we can find a counterexample to the first part

$$A \to B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha]B$$

If yes, we disproved (5), otherwise we use [*] once more.

This process of iteratively unrolling a loop with either axiom [*] or rule $[*^n]$r is called *Bounded Model Checking* and has been used very successfully, e.g., in the context of finite-state systems [CBRZ01]. The same principle can be useful to disprove properties of loops in differential dynamic logic by unwinding the loop.

5 Breaking Loops for Proofs

Proving properties of loops by unwinding them forever with $[*^n]$r is not a promising strategy, unless we find that the conjecture is not valid after a number of unwindings. One way or another, we will have to find a way to break the loop apart to complete our reasoning.

Consider the formula (11) again that we got from (5) by unwinding the loop with axiom [*] a number of times and then flattening the formula with the help of []∧:

$$A \to B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha]B \land [\alpha][\alpha][\alpha^*]B \tag{11}$$

Using →r and ∧r on (11) leads to

$$\to\text{r}\,\dfrac{\land\text{r}\,\dfrac{A \vdash B \qquad \land\text{r}\,\dfrac{A \vdash [\alpha]B \qquad \land\text{r}\,\dfrac{A \vdash [\alpha][\alpha]B \qquad \land\text{r}\,\dfrac{A \vdash [\alpha][\alpha][\alpha]B \qquad A \vdash [\alpha][\alpha][\alpha^*]B}{A \vdash [\alpha][\alpha][\alpha]B \land [\alpha][\alpha][\alpha^*]B}}{A \vdash [\alpha][\alpha]B \land [\alpha][\alpha][\alpha]B \land [\alpha][\alpha][\alpha^*]B}}{A \vdash [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha]B \land [\alpha][\alpha][\alpha^*]B}}{A \vdash B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha]B \land [\alpha][\alpha][\alpha^*]B}}{\vdash A \to B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha]B \land [\alpha][\alpha][\alpha^*]B}$$

Let us summarize this notationally by the following

$$\to\text{r},\land\text{r},\land\text{r},\land\text{r},\land\text{r}\,\dfrac{A \vdash B \quad A \vdash [\alpha]B \quad A \vdash [\alpha][\alpha]B \quad A \vdash [\alpha][\alpha][\alpha]B \quad A \vdash [\alpha][\alpha][\alpha^*]B}{\vdash A \to B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha]B \land [\alpha][\alpha][\alpha^*]B}$$

to recall that there was a derivation involving one use of →r and 4 uses of ∧r from the four premises to the single conclusion without saying which derivation it was exactly. Mentioning ∧r 4 times seems a bit repetitive, so simply abbreviate this as:

$$\to\text{r},\land\text{r}\,\dfrac{A \vdash B \quad A \vdash [\alpha]B \quad A \vdash [\alpha][\alpha]B \quad A \vdash [\alpha][\alpha][\alpha]B \quad A \vdash [\alpha][\alpha][\alpha^*]B}{\vdash A \to B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha]B \land [\alpha][\alpha][\alpha^*]B}$$

How could we prove the premises? Sect. 4 investigated one way, which essentially amounts to Bounded Model Checking. Can we be more clever and prove the same premises in a different way? Preferably one that is more efficient?

There is to much we can do to improve the way we prove the first premise. We simply have to bite the bullet and do it, armed with all our knowledge of arithmetic. But it's actually very easy at least for the bouncing ball. Besides, no dynamics has actually happened yet in the first premise, so if we despair in proving this one, the rest cannot become any easier either. For the second premise, there is not much that we can do, because we will have to analyze the effect of the loop body α running once at least in order to be able to understand what happens if we run α repeatedly.

Yet, what's with the third premise $A \vdash [\alpha][\alpha]B$? We could just approach it as is and try to prove it directly using the d$\mathcal{L}$ proof rules. Alternatively, however, we could try to take advantage of the fact that it is the same hybrid program α that is running in the first and the second modality. Maybe they should have something in common that we can exploit as part of our proof?

How could that work? Can we possibly find something that the is true after the first run of α and is all we need to know about the state for $[\alpha]B$ to hold? Can we characterize the intermediate state after the first α and before the second α? Suppose we manage to do that and identify a formula E that characterizes the intermediate state in this way. How do we use intermediate condition E to simplify our proof?

Recall the intermediate condition contract version of the sequential composition proof rule from Lecture 4 and Lecture 5.

$$\text{(R4)}\ \frac{A \to [\alpha]E \quad E \to [\beta]B}{A \to [\alpha;\beta]B}$$

Lecture 5 ended up dismissing the intermediate contract rule R4 in favor of the more general axiom

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

But, let us revisit R4 and see if we can learn something from its way of using intermediate condition E. The first obstacle is that the conclusion of R4 does not match the form we need for $A \vdash [\alpha][\alpha]B$. That's not a problem in principle, because we could use axiom [;] backwards from right-hand side to left-hand side in order to turn $A \vdash [\alpha][\alpha]B$ into

$$A \vdash [\alpha;\alpha]B$$

and then use rule R4. However, this is what we wanted to stay away from, because using the axioms both forwards and backwards can get our proof search into trouble because we might loop around trying to find a proof forever without making any progress by simply using [;] forwards and then backwards and then forwards again and so on until the end of time. That does not strike us as useful. Instead, we'll adopt a proof rule that has some of the thoughts of R4 but is more general. It is called *generalization*:

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

Rule $[]gen'$ on the third premise $A \vdash [\alpha][\alpha]B$ with the intermediate condition E for ϕ that we assume to have identified

$$[]gen'\,\dfrac{A \vdash [\alpha]E \qquad E \vdash [\alpha]B}{A \vdash [\alpha][\alpha]B}$$

Let us try to use this principle to see if we can find a way to prove

$$A \rightarrow B \land [\alpha](B \land [\alpha](B \land [\alpha](B \land [\alpha][\alpha^*]B))) \tag{9}$$

Using $\land$r and $[]gen'$ a number of times for a sequence of intermediate conditions E_1, E_2, E_3 derives:

$$\rightarrow\text{r}\,\dfrac{\land\text{r}\,\dfrac{A \vdash B \quad []gen'\,\dfrac{A \vdash [\alpha]E_1 \quad \land\text{r}\,\dfrac{E_1 \vdash B \quad []gen'\,\dfrac{E_1 \vdash [\alpha]E_2 \quad \land\text{r}\,\dfrac{E_2 \vdash B \quad []gen'\,\dfrac{E_2 \vdash [\alpha]E_3 \quad \land\text{r}\,\dfrac{E_3 \vdash B \quad E_3 \vdash [\alpha][\alpha^*]B}{E_3 \vdash B \land [\alpha][\alpha^*]B}}{E_2 \vdash [\alpha](B \land [\alpha][\alpha^*]B)}}{E_2 \vdash B \land [\alpha](B \land [\alpha][\alpha^*]B)}}{E_1 \vdash [\alpha](B \land [\alpha](B \land [\alpha][\alpha^*]B))}}{E_1 \vdash B \land [\alpha](B \land [\alpha](B \land [\alpha][\alpha^*]B))}}{A \vdash [\alpha](B \land [\alpha](B \land [\alpha](B \land [\alpha][\alpha^*]B)))}}{A \vdash B \land [\alpha](B \land [\alpha](B \land [\alpha](B \land [\alpha][\alpha^*]B)))}}{\vdash A \rightarrow B \land [\alpha](B \land [\alpha](B \land [\alpha](B \land [\alpha][\alpha^*]B)))}$$

This particular derivation is still not very useful because it still has a loop in one of the premises, which is what we had originally started out with in (5) in the first place. But the derivation hints at a useful way how we could possibly shortcut proofs. To lead to a proof of the conclusion, the above derivation requires us to prove the premises

$$A \vdash [\alpha]E_1$$
$$E_1 \vdash [\alpha]E_2$$
$$E_2 \vdash [\alpha]E_3$$

as well as some other premises. What if all the intermediate conditions E_i were the same? Let's assume they are all the same condition E, that is, $E_1 \equiv E_2 \equiv E_3 \equiv E$. Then most of the premises turn out to be the same:

$$E \vdash B$$
$$E \vdash [\alpha]E$$

except for the two left-most and the right-most premise. Let us leverage this observation and develop a proof rule for which the same intermediate condition is used for all iterates of the loop. Furthermore, we would even know the first premise

$$A \vdash [\alpha]E$$

if we could prove that the precondition A implies E:

$$A \vdash E$$

because, we already have $E \vdash [\alpha]E$.

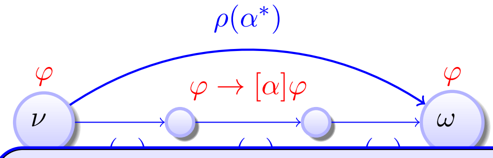
6 Invariant Proofs of Loops

The condition $E \vdash [\alpha]E$ identified in the previous section seems particularly useful, because it basically says that whenever the system α starts in a state satisfying E, it will stay in E. It sounds like the system α^* couldn't get out of E either if it starts in E since all that α^* can do is to repeat α some number of times. But every time we repeat α, the sequent $E \vdash [\alpha]E$ expresses that we cannot leave E that way.

The other condition that the previous section identified as crucial is $E \vdash B$. And, indeed, if E does not imply the postcondition B that we have been interested in in the first place, then E is not necessarily very useful to prove B.

Recall from Lecture 3

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



Lemma 2 (Induction). *The induction rule is sound:*

$$(ind')\ \frac{\Gamma \vdash \varphi, \Delta \quad \varphi \vdash [\alpha]\varphi \quad \varphi \vdash \psi}{\Gamma \vdash [\alpha^*]\psi, \Delta}$$

First observe that the *inductive invariant* φ (which we called E in the previous examples) occurs in all premises but not in the conclusion of *ind′*. The first premise of *ind′* says that the initial state, about which we assume Γ (and that Δ does not hold), satisfies the invariant φ. The second premise of *ind′* shows that the invariant φ is inductive. That is, whenever φ was true before running the loop body α, then φ is always true again after running α. The third premise of *ind′* shows that the invariant φ is strong enough to imply the postcondition ψ that the conclusion was interested in.

Rule *ind′* says that ψ holds after any number of repetitions of α if an invariant φ holds initially (left premise) and invariant φ remains true after one iteration of α (middle premise), and invariant φ finally implies the desired postcondition ψ (right premise). If φ is true after executing α whenever φ has been true before (middle premise), then, if φ holds in the beginning (left premise), φ will continue to hold, no matter how often we repeat α in $[\alpha^*]\psi$, which is enough to imply $[\alpha^*]\psi$ if φ implies ψ.

Taking a step back, these three premises correspond exactly to the proof steps that 15-122 Principles of Imperative Computation used to show that the contract of a function with a `@requires` contract Γ (and not Δ), `@ensures` contract ψ, and a loop invariant φ is correct. Now, we have this reasoning in a more general and formally more precisely defined context.

7 A Proof of a Repetitive Bouncing Ball

$$
\begin{aligned}
&\texttt{@requires}(0 \leq h \wedge h = H \wedge v = 0)\\
&\texttt{@requires}(g > 0 \wedge 1 \geq c \geq 0)\\
&\texttt{@ensures}(0 \leq h \wedge h \leq H)\\
&\big(h' = v, v' = -g \,\&\, h \geq 0;\\
&\quad (?h = 0; v := -cv \cup ?h \geq 0)\big)\big)^{*}\texttt{@invariant}(2gh = 2gH - v^2 \wedge h \geq 0)
\end{aligned}
\tag{12}
$$

Let us again use abbreviations:

$$
\begin{aligned}
A_{h,v} &\overset{\text{def}}{\equiv} 0 \leq h \wedge h = H \wedge v = 0 \wedge g > 0 \wedge 1 \geq c \geq 0\\
B_{h,v} &\overset{\text{def}}{\equiv} 0 \leq h \wedge h \leq H\\
(h'' = \dots) &\overset{\text{def}}{\equiv} (h' = v, v' = -g \,\&\, h \geq 0)\\
E_{h,v} &\overset{\text{def}}{\equiv} 2gh = 2gH - v^2 \wedge h \geq 0
\end{aligned}
$$

Note the somewhat odd abbreviation for the differential equation just to simplify notation. Also note the invariant $E_{h,v}$ that we identified as an intermediate condition for the single-hop bouncing ball in Lecture 4 on Safety & Contracts. After the considerations in Sect. 5, it should no longer be a big surprise why we try to use an intermediate condition as an invariant. We are not sure whether this will work but it seems worth trying.

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

Let there be proof.

$$
{\scriptstyle \rightarrow\mathrm{r}}\,\dfrac{
{\scriptstyle ind'}\,\dfrac{
A_{h,v} \vdash E_{h,v}
\quad
{\scriptstyle [;]\mathrm{r}}\,\dfrac{
{\scriptstyle []gen'}\,\dfrac{
E_{h,v} \vdash [h' = \dots]E_{h,v}
\quad
{\scriptstyle [\cup]\mathrm{r}}\,\dfrac{
{\scriptstyle \wedge\mathrm{r}}\,\dfrac{
{\scriptstyle [;]\mathrm{r}}\,\dfrac{
{\scriptstyle [?]\mathrm{r}}\,\dfrac{
{\scriptstyle [:=]\mathrm{r}}\,\dfrac{E_{h,v}, h = 0 \vdash E_{h,-cv}}{E_{h,v}, h = 0 \vdash [v := -cv]E_{h,v}}
}{E_{h,v} \vdash [?h = 0][v := -cv]E_{h,v}}
}{E_{h,v} \vdash [?h = 0; v := -cv]E_{h,v}}
\quad
{\scriptstyle [?]\mathrm{r}}\,\dfrac{E_{h,v}, h \geq 0 \vdash E_{h,v}}{E_{h,v} \vdash [?h \geq 0]E_{h,v}}
}{E_{h,v} \vdash [?h = 0; v := -cv]E_{h,v} \wedge [?h \geq 0]E_{h,v}}
}{E_{h,v} \vdash [?h = 0; v := -cv \cup ?h \geq 0]E_{h,v}}
}{E_{h,v} \vdash [h' = \dots][?h = 0; v := -cv \cup ?h \geq 0]E_{h,v}}
}{E_{h,v} \vdash [h' = \dots; (?h = 0; v := -cv \cup ?h \geq 0]E_{h,v}}
\quad
E_{h,v} \vdash B_{h,v}
}{A_{h,v} \vdash [(h' = \dots; (?h = 0; v := -cv \cup ?h \geq 0)^{*}]B_{h,v}}
}{\vdash A_{h,v} \rightarrow [(h' = \dots; (?h = 0; v := -cv \cup ?h \geq 0)^{*}]B_{h,v}}
$$

The remaining 5 premises are prove easily. The first premise $A_{h,v} \vdash E_{h,v}$ proves easily using $h = H$ and $v = 0$:

$$
0 \leq h \wedge h = H \wedge v = 0 \wedge g > 0 \wedge 1 \geq c \geq 0 \vdash 2gh = 2gH - v^2 \wedge h \geq 0
$$

Recalling the unusual abbreviations, the second premise $E_{h,v} \vdash [h' = \dots]E_{h,v}$ is

$$
2gh = 2gH - v^2 \wedge h \geq 0 \vdash [h' = v, v' = -g \,\&\, h \geq 0](2gh = 2gH - v^2 \wedge h \geq 0)
$$

a proof whose pieces we have seen in previous lectures (Exercise 2). The third premise $E_{h,v}, h = 0 \vdash E_{h,-cv}$ is

$$2gh = 2gH - v^2 \land h \geq 0, h = 0 \vdash 2gh = 2gH - (-cv)^2 \land h \geq 0$$

which would prove easily if we knew $c = 1$. Do we know $c = 1$? No we do not know $c = 1$, because we only assumed $1 \geq c \geq 0$ in $A_{h,v}$. But we could prove this third premise easily if we would change the definition of $A_{h,v}$ around to include $c = 1$. Note that even then, however, we still need to augment $E_{h,v}$ to include $c = 1$ as well, since we otherwise would have lost this knowledge before we need it in the third premise. The fourth premise, $E_{h,v}, h \geq 0 \vdash E_{h,v}$ proves whatever the abbreviations stand for simply using the axiom rule ax. Finally, the fifth premise $E_{h,v} \vdash B_{h,v}$, which is

$$2gh = 2gH - v^2 \land h \geq 0 \vdash 0 \leq h \land h \leq H$$

proves easily with arithmetic as long as we know $g > 0$. This condition is already included in $A_{h,v}$. But we still managed to forget about that in our intermediate condition. So, again, $g > 0$ should have been included in the invariant $E_{h,v}$, which should have been defined as

$$E_{h,v} \overset{\text{def}}{\equiv} 2gh = 2gH - v^2 \land h \geq 0 \land c = 1 \land g > 0$$

Yet, only the last two conjuncts are trivial, because neither c nor g changes while the little bouncing ball falls. We, unfortunately, still have to include it in the invariant. This is one of the downsides of working with intermediate condition style proofs such as what we get with rule $[]gen'$. Later lectures investigate significant simplifications for this nuisance.

For the record, we now have a sequent proof of the undamped bouncing ball with repetitions:

$$\begin{aligned}&0 \leq h \land h = H \land v = 0 \land g > 0 \land 1 = c \rightarrow \\ &\qquad [\big(h' = v, v' = -g \,\&\, h \geq 0; (?h = 0; v := -cv \cup ?h \geq 0)\big)^*](0 \leq h \land h \leq H)\end{aligned} \tag{13}$$

Looking back, the contract in (12) has almost reflected this, but not quite, because the `@invariant` contract forgot to capture the constant invariants $c = 1 \land g > 0$. And the `@requires` contract forgot to require $c = 1$. Let's capture this contract, which we have now verified by way of proving the corresponding d$\mathcal{L}$ formula (13):

$$\begin{aligned}&\texttt{@requires}(0 \leq h \land h = H \land v = 0)\\ &\texttt{@requires}(g > 0 \land c = 1)\\ &\texttt{@ensures}(0 \leq h \land h \leq H)\\ &\big(h' = v, v' = -g \,\&\, h \geq 0;\\ &\quad (?h = 0; v := -cv \cup ?h \geq 0))\big)^* \texttt{@invariant}(2gh = 2gH - v^2 \land h \geq 0 \land c = 1 \land g > 0)\end{aligned} \tag{14}$$

8 Essentials of Induction & Cuts

The induction rule *ind′* is very useful in practice. But there is a more elegant and more essential way of stating the induction principle.

Lemma 3 (Induction)**.** *The induction rule is sound:*

$$(\mathit{ind})\ \frac{\varphi \vdash [\alpha]\varphi}{\varphi \vdash [\alpha^*]\varphi}$$

ind is clearly a special case of *ind′*, obtained by specializing $\Gamma \overset{\text{def}}{\equiv} \psi\ \Delta = .$, and $\varphi \overset{\text{def}}{\equiv} \psi$, in which case the left and right premises of *ind′* are provable directly by *ax* so that only the middle premise remains. If *ind* is a special case of *ind′*, why should we still prefer *ind* from a perspective of essentials? Obviously, *ind* is more fundamental and easier. But if this came at the cost of being less powerful, *ind′* should still be preferred. It turns out that *ind′* is actually a special case of *ind* with a little extra work. This extra work needs a bit of attention but is insightful.

Let's adopt the following variation of the generalization rule:

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

For example, using a cut with $\varphi \to [\alpha^*]\varphi$, rule *ind′* can be derived from *ind* and $[]\mathit{gen}$ as follows (using weakening Wl,Wr without notice):

$$\mathit{cut}\,\dfrac{{\to}\mathrm{r}\,\dfrac{\mathit{ind}\,\dfrac{\varphi \vdash [\alpha]\varphi}{\varphi \vdash [\alpha^*]\varphi}}{\Gamma \vdash \varphi \to [\alpha^*]\varphi, \Delta} \qquad {\to}\mathrm{l}\,\dfrac{\Gamma \vdash \varphi, \Delta \qquad []\mathit{gen}\,\dfrac{\varphi \vdash \psi}{[\alpha^*]\varphi \vdash [\alpha^*]\psi}}{\Gamma, \varphi \to [\alpha^*]\varphi \vdash [\alpha^*]\psi, \Delta}}{\Gamma \vdash [\alpha^*]\psi, \Delta}$$

Hence *ind′* is a derived rule, because it can be derived using *ind* and some other rules. Thus, *ind′* is not necessary in theory, but still useful in practice.

Yet, now, in order to derive rule *ind′* out of the more fundamental *ind*, we had to add the revised generalization rule $[]\mathit{gen}$. Is that any easier? Well it is, because $[]\mathit{gen}$ actually makes $[]\mathit{gen}'$ unnecessary by another smart argument using a *cut* with the desired formula $[\alpha]\phi$.

$$\mathit{cut}\,\dfrac{\mathrm{Wr}\,\dfrac{\Gamma \vdash [\alpha]\phi, \Delta}{\Gamma \vdash [\alpha]\phi, [\alpha]\psi, \Delta} \qquad \mathrm{Wl,Wr}\,\dfrac{[]\mathit{gen}\,\dfrac{\phi \vdash \psi}{[\alpha]\phi \vdash [\alpha]\psi}}{\Gamma, [\alpha]\phi \vdash [\alpha]\psi, \Delta}}{\Gamma \vdash [\alpha]\psi, \Delta}$$

This leaves exactly the premises of rule $[]\mathit{gen}'$, making $[]\mathit{gen}'$ a derived rule. Whenever we need $[]\mathit{gen}'$, we could simply expand the proof out in the above form to reduce it just a proof involving $[]\mathit{gen}$ and *cut* and weakening.

These are two illustrations how creative uses of cuts can suddenly make proves and concepts easier. A phenomenon that we will see in action much more often in this course.

Before you despair that you would have to derive ind' and $[]gen'$ every time you need them: that is not the case. The theorem prover KeYmaera is very well aware of how useful both versions of the proof rules are and has them at your disposal. For theoretical investigations, however, as well as for understanding the truly fundamental reasoning steps, it is instructive to see that ind and $[]gen$ are fundamental, while the others are mere consequences.

Exercises

Exercise 1 (Give bouncing ball back its evolution domain). Explain why the transformation from (3) to (4) was okay in this case.

Exercise 2. Give a sequent proof for

$$2gh = 2gH - v^2 \land h \geq 0 \rightarrow [h' = v, v' = -g \,\&\, h \geq 0](2gh = 2gH - v^2 \land h \geq 0)$$

Does this property also hold if we remove the evolution domain constraint $h \geq 0$? That is, is the following formula valid?

$$2gh = 2gH - v^2 \land h \geq 0 \rightarrow [h' = v, v' = -g](2gh = 2gH - v^2 \land h \geq 0)$$

Exercise 3. To develop an inductive proof rule, we have started systematic unwinding considerations from formula (9) in Sect. 5. In lecture, we started from the form (11) instead and have seen that that takes us to the same inductive principle. Which of the two ways of proceeding is more efficient? Which one produces less premises that are distractions in the argument? Which one has less choices of different intermediate conditions E_i in the first place?

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.

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

[Pla12b] André Platzer. Logics of dynamical systems. In *LICS*, pages 13–24. IEEE, 2012. doi:10.1109/LICS.2012.13.