Chapter 7
Control Loops & Invariants

Synopsis This chapter advances the analytical understanding of cyber-physical systems to cover control loops. While the syntax and semantics of hybrid programs from previous chapters already discussed loops, their logical characterization was so far limited to unfolding by the iteration axiom. That suffices for systems with a fixed finite number of control actions in a fixed finite number of repetitions of the control loop, but is not enough to understand and analyze the most interesting CPSs with unbounded time-horizons reaching an unbounded number of control decisions over time. This chapter uses the fundamental concept of invariants to handle loops and develops their operational intuition. CPS invariants are developed systematically based on inductive formulations of dynamic axioms for repetitions.

7.1 Introduction

Chap. 5 introduced rigorous reasoning for hybrid program models of cyber-physical systems, which Chap. 6 extended to a systematic and coherent reasoning approach for cyber-physical systems. Our understanding of the language exceeds our understanding of the reasoning principles, though, because we have not seen any credible ways of analyzing loops yet, despite the fact that loops are a perfectly harmless and common part of CPSs. In fact, computational thinking would argue that we do not truly understand an element of a programming language or a system model if we do not also understand ways of reasoning about them. This chapter sets out to make sure our analysis capabilities catch with on our modeling skills. This is, of course, all part of the agenda we set forth initially to study the language of cyber-physical systems gradually in layers that we master completely before advancing to the next challenge. The next challenge is control loops.

Chap. 3 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 to get our CPS anywhere. With loops, CPS controllers shine, because they

A. Platzer, *Logical Foundations of Cyber-Physical Systems*,
https://doi.org/10.1007/978-3-319-63588-0_7

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. Loops truly make feedback happen, by enabling a CPS to sense state and act in response to that over and over again. Think of programming a robot to drive on a highway. Would you be able to do that without some means of repetition or iteration as in repeated control? Probably not, because you would 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. There's no way of telling ahead of time, how often the robot will need to change its mind when it's driving a car on a highway.

A hybrid program's way of exercising repetitive control actions is the repetition operator *, which can be applied to any hybrid program α. The resulting hybrid program α^* repeats α any number of times, nondeterministically. That may be zero times or one time or 10 times or ....

Now, the flip side of the fact that control loops are responsible for a lot of the power of CPS is that they can also be tricky to analyze and fully understand. After all, it is easier to get a handle on what a system does in just one step than to understand what it will do in the long run when the CPS is running for any arbitrary amount of time. This is the CPS analogue of the fact that ultra-short-term predictions are often much easier than long-term predictions. It is easy to predict the weather a second into the future but much harder to predict next week's weather.[1]

The main insight behind the analysis of loops in CPS is to reduce the (complicated) analysis of their long-term global behavior to a simpler analysis of their local behavior for one control cycle. This principle significantly reduces the analytic complexity of loops in CPS. It leverages invariants, i.e., aspects of the system behavior that do not change as time progresses, so that our analysis can rely on them no matter how long the system already evolved. Invariants turn out also to lead to an important design principle for CPS, even more so than in programs. The significance of invariants in understanding CPS is not a coincidence, because the study of invariants (as with other mathematical structures) is also central to a large body of mathematics.

Since it is of central importance to develop a sense of how the parts of a proof fit together and what impact changes to preconditions or invariants have on a proof, this chapter will be very explicit about developing sequent calculus proofs to give you a chance to understand their structure. These proofs will also serve as a useful exercise to practice our skills on the sequent calculus reasoning for CPS that Chap. 6 developed. After some practice, subsequent chapters will often appeal in more intuitive ways to the canonical structure that a proof will have and focus on developing only its most crucial elements: invariants, because the remaining proof is relatively straightforward.

The most important learning goals of this chapter are:

[1] Of course, Nils Bohr already figured this out when he said that "prediction is very difficult, especially if it's about the future."

Modeling and Control: We develop a deeper understanding of control loops as a core principle behind CPS that ultimately underlies all feedback mechanisms in CPS control. This chapter also intensifyies our understanding of the dynamical aspects of CPS and how discrete and continuous dynamics interact.

Computational Thinking: This chapter extends the rigorous reasoning approach from Chap. 5 to systems with repetitions. This chapter is devoted to the development of rigorous reasoning techniques for CPS models with repetitive control loops or other loopy behavior, a substantially nontrivial problem in theory and practice. Without understanding loops, there is no hope of understanding the repetitive behavior of feedback control principles that are common to almost all CPSs. Understanding such behavior can be tricky, because so many things can change in the system and its environment over the course of the runtime of even just a few lines of code if that program runs repeatedly to control the behavior of a CPS. That is why the study of *invariants*, i.e., properties that do not change throughout the execution of the system, are crucial for their analysis. Invariants constitute the single most insightful and most important piece of information about a CPS. As soon as we understand the invariants of a CPS, we almost understand everything about it and will even be in a position to design the rest of the CPS around these invariants, a process known as the design-by-invariant principle. Identifying and expressing invariants of CPS models will be a part of this chapter as well.

The first part of the chapter shows a systematic development of invariance principles for loops from an axiomatic basis. The second part of the chapter focuses on loop invariants themselves along with their operational intuition.

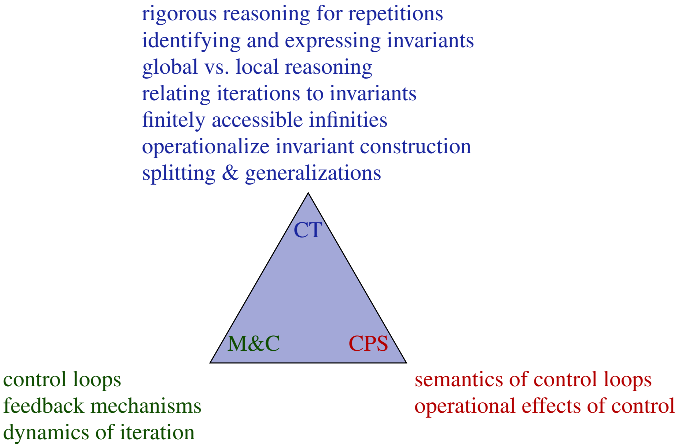
Another aspect that this chapter reinforces is the important concept of global proof rules, which, just like Gödel's generalization rule G, for soundness reasons cannot keep the sequent context.

CPS Skills: We will develop a better understanding of the semantics of CPS models by understanding the core aspects of repetition and relating its semantics to corresponding reasoning principles. This understanding will lead us to develop a higher level of intuition for the operational effects involved in CPS by truly understanding what control loops fundamentally amount to.

7.2 Control Loops

Recall Quantum, the little acrophobic bouncing ball from Chap. 4:

$$\begin{aligned}
&\mathbf{requires}(0 \le x \land x = H \land v = 0)\\
&\mathbf{requires}(g > 0 \land 1 \ge c \ge 0)\\
&\mathbf{ensures}(0 \le x \land x \le H)\\
&\big(\{x' = v, v' = -g \,\&\, x \ge 0\};\\
&\;\;\mathsf{if}(x = 0)\, v := -cv\big)^*
\end{aligned} \tag{4.25*}$$


rigorous reasoning for repetitions
identifying and expressing invariants
global vs. local reasoning
relating iterations to invariants
finitely accessible infinities
operationalize invariant construction
splitting & generalizations
CT
M&C
CPS
control loops
feedback mechanisms
dynamics of iteration
semantics of control loops
operational effects of control

The contracts above have been augmented with the ones that we have identified in Chap. 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:

$$0 \leq x \land x = H \land v = 0 \land g > 0 \land 1 \geq c \geq 0 \rightarrow \\ \big[\big(\{x' = v, v' = -g \,\&\, x \geq 0\};\ \mathsf{if}(x = 0)\, v := -cv\big)^*\big]\,(0 \leq x \land x \leq H) \quad (4.23^*)$$

As we do not wish to be bothered by the presence of the additional if-then-else operator, which is not officially part of the minimal set of operators that differential dynamic logic dL provides, we rewrite (4.23) equivalently to:

$$0 \leq x \land x = H \land v = 0 \land g > 0 \land 1 \geq c \geq 0 \rightarrow \\ \big[\big(\{x' = v, v' = -g \,\&\, x \geq 0\};\ (?x = 0; v := -cv \cup ?x \neq 0)\big)^*\big]\,(0 \leq x \land x \leq H) \quad (7.1)$$

In Chap. 4, we had an informal understanding why (7.1) is valid (true in all states), but no formal proof, albeit we proved a much simplified version of (7.1) in which we simply threw away the loop. Such ignorance is clearly not a correct way of understanding loops. Equipped with our refined understanding of what proofs are from Chap. 6, let's make up for that now by properly proving (7.1) in the dL calculus.

However, before going for a proof of this bouncing-ball property, however much Quantum may long for it, let us first take a step back and understand the rôle of loops in more general terms. Their semantics has been explored in Chap. 3 with unwinding-based reasoning in Chap. 5.

Quantum had a loop in which physics and its bouncing control alternated. Quantum desperately needs a loop for he doesn't know ahead of time how often he would bounce today. When falling from a great height, Quantum bounces quite a bit. Quantum also had a controller, albeit a rather impoverished one. All it can do is inspect the current height, compare it to the ground floor (at height 0) and, if $x = 0$, flip its velocity vector around after some casual damping by factor c. That is not a whole lot of flexibility for control choices, but Quantum was still rather proud to serve such an important rôle in controlling the ball's behavior. Indeed, without the control action, Quantum would never bounce back from the ground but would keep on falling forever—what a frightful thought for the acrophobic Quantum. On second thought Quantum would, actually, not even fall for very long without its controller, because of the evolution domain $x \geq 0$ for physics $x'' = -g \,\&\, x \geq 0$, which only allows 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 the $x \geq 0$ constraint won't have it. So, in summary, without Quantum'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 Quantum is really jolly proud of the controller.

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 for it in the first place. Hence, control is crucial and understanding and analyzing its effect on physics is one of the primary responsibilities in CPS. After the implication in (7.1) is quickly consumed by the →R proof rule, the trouble starts right away since Quantum needs to prove the safety of the loop.

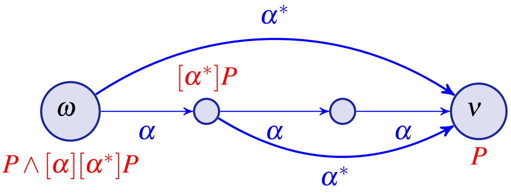
7.3 Induction for Loops

This section develops induction principles for loops by systematically developing their intuition starting from the insights behind the iteration axiom.

7.3.1 Induction Axiom for Loops

Recall the loop semantics from Sect. 3.3.2 and its unwinding axiom from Sect. 5.3.7:

$$[\![\alpha^*]\!] = [\![\alpha]\!]^* = \bigcup_{n \in \mathbb{N}} [\![\alpha^n]\!] \qquad \text{with} \quad \alpha^{n+1} \equiv \alpha^n; \alpha \text{ and } \alpha^0 \equiv ?\mathit{true}$$



Lemma 5.7 (**[∗] iteration axiom**). *The* iteration axiom *is sound:*

$$[^*] \quad [\alpha^*]P \leftrightarrow P \land [\alpha][\alpha^*]P$$

Using the iteration axiom [∗] from left to right, it "reduces" a safety property

$$[\alpha^*]P \tag{7.2}$$

of a loop α^* to the following equivalent dL formula:

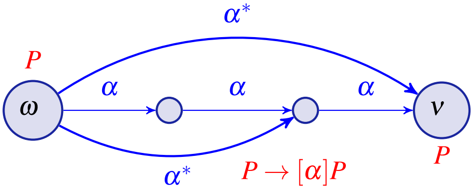
$$P \land [\alpha][\alpha^*]P \tag{7.3}$$

The isolated left formula P and the $[\alpha]$ modality in the resulting formula (7.3) are simpler than the original (7.2) and could, thus, be analyzed using the other dL axioms. The only catch is that the postcondition $[\alpha^*]P$ of the $[\alpha]$ modality in (7.3) is as complicated as the original dL formula (7.2). While the iteration axiom [∗] unpacked necessary conditions for the original repetition property (7.2), the true question of whether P always holds after repeating α any number of times remains, albeit nested within an extra $[\alpha]$. That does not look like a lot of progress in analyzing (7.2). In fact, it looks like using the iteration axiom [∗] makes matters more complicated (unless perhaps a counterexample has been identified along the way). The iteration axiom [∗] can still be useful to explicitly uncover the effect of one round of a loop.

Since (7.2) and (7.3) are equivalent, formula $[\alpha^*]P$ can only be true if P holds initially. So, if, in some state ω, we are trying to establish $\omega \in [\![[\alpha^*]P]\!]$, then we only have a chance if the necessary condition $\omega \in [\![P]\!]$ holds in the initial state ω. By the equivalent (7.3), $\omega \in [\![[\alpha^*]P]\!]$ can also only hold if $\omega \in [\![[\alpha]P]\!]$ since the loop in $[\alpha][\alpha^*]P$ may repeat 0 times (Exercise 7.2). So, we might as well establish the necessary condition $\omega \in [\![P \to [\alpha]P]\!]$ since we already needed to assume $\omega \in [\![P]\!]$. Showing the implication $P \to [\alpha]P$ in state ω is a little easier than showing $[\alpha]P$, because the implication assumes P. This shows $\mu \in [\![P]\!]$ in any state μ after the *first* loop iteration, but since its α-successors will all also have to satisfy P for $\omega \in [\![[\alpha^*]P]\!]$ to hold, we again need to show the same remaining condition $P \to [\alpha]P$, just in a different state μ.

If, instead, we manage to prove $P \to [\alpha]P$ *in all states* we get to by repeating α, not just the initial state ω, then we know P holds in all states after running α twice from ω, since we already know that P holds in all states μ after running α once from ω. By induction, no matter how often α is repeated, we know P is true afterwards if only P was true initially and $P \to [\alpha]P$ is always true after repeating α, i.e., $[\alpha^*](P \to [\alpha]P)$ is true in the current state, which is $\omega \in [\![[\alpha^*](P \to [\alpha]P)]\!]$.

These thoughts lead to the induction axiom I expressing that a property P is always true after repeating HP α iff P is true initially and if, after any number of repetitions of α, P always holds after one more repetition of α if it held before.



Lemma 7.1 (I induction axiom). *The induction axiom is sound:*

$$\text{I} \quad [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P)$$

Proof. Let $\omega \in [\![[\alpha^*]P]\!]$, then $\omega \in [\![P]\!]$ by choosing 0 iterations and $\omega \in [\![[\alpha^*][\alpha]P]\!]$ by choosing at least one iteration, which implies $\omega \in [\![[\alpha^*](P \rightarrow [\alpha]P)]\!]$. Conversely, let $\omega \in [\![P \land [\alpha^*](P \rightarrow [\alpha]P)]\!]$. Then consider a run of α^* from ω to ν with $n \in \mathbb{N}$ iterations, i.e., $(\omega, \nu) \in [\![\alpha^n]\!]$. The proof shows $\nu \in [\![P]\!]$ by induction on n (Fig. 7.1).

0. Case $n = 0$: Then $\nu = \omega$ satisfies $\nu \in [\![P]\!]$ by the first conjunct.
1. Case $n+1$: By induction hypothesis for n, all states μ with $(\omega, \mu) \in [\![\alpha^n]\!]$ are assumed to satisfy $\mu \in [\![P]\!]$. Thus, $\mu \in [\![[\alpha]P]\!]$ by the second conjunct $\omega \in [\![[\alpha^*](P \rightarrow [\alpha]P)]\!]$ since $(\omega, \mu) \in [\![\alpha^n]\!] \subseteq [\![\alpha^*]\!]$. Hence, $\nu \in [\![P]\!]$ for all states ν with $(\mu, \nu) \in [\![\alpha]\!]$. Thus, $\nu \in [\![P]\!]$ for all states ν with $(\omega, \nu) \in [\![\alpha^{n+1}]\!]$. □

The $[\alpha^*]$ modality on the right-hand side of axiom I is necessary for soundness, because it would not be enough to merely show that the implication $P \rightarrow [\alpha]P$ is true in the current state. That is, the following formula would be an unsound axiom

$$[\alpha^*]P \leftrightarrow P \land (P \rightarrow [\alpha]P)$$

because its instance

$$[(x:=x+1)^*]x \leq 2 \leftrightarrow x \leq 2 \land (x \leq 2 \rightarrow [x:=x+1]x \leq 2)$$

is not true in a state ω with $\omega(x) = 0$, so it is also not valid. The $[\alpha^*]$ modality on the right-hand side of axiom I ensures that $P \rightarrow [\alpha]P$ is not just true in the current state, but true in all states reached after iterating the loop α^* any number of times.

7.3.2 Induction Rule for Loops

Even if axiom I has a pleasantly inductive flair to it, using it directly does not make matters any better compared to the iteration axiom [*]. Using axiom I to prove a

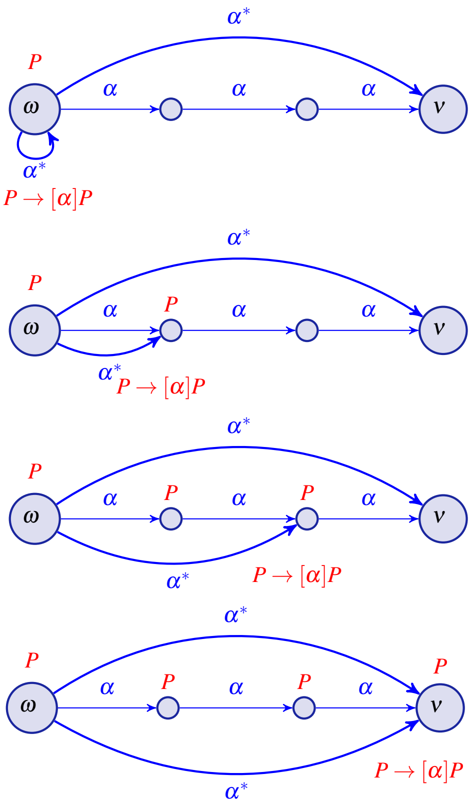


Fig. 7.1 Successively using induction axiom I at each state reached after running iterations of α^*

property of a loop (its left-hand side) will still reduce to proving a different property of a loop (its right-hand side). Why should the postcondition $P \to [\alpha]P$ be any easier to prove after the loop α^* than the original postcondition P?

The clou, however, is that the postcondition $P \to [\alpha]P$ in the right-hand side of induction axiom I can also be proved differently. Gödel's generalization rule G, which was already discussed in Sect. 5.6.3, provides a way of proving postconditions of arbitrary box modalities, even with loops, if the postcondition has a proof. Valid formulas (premise) are also true after all runs of any HP α (conclusion).

Lemma 5.12 (G Gödel generalization rule). *The* Gödel rule *is sound:*

$$\text{G}\ \frac{P}{[\alpha]P}$$

Generalization rule G can be used to prove $[\alpha^*](P \to [\alpha]P)$ by proving the postcondition $P \to [\alpha]P$. This leads to the induction rule, which reduces the proof that P always holds after repeating α (succedent of conclusion) provided that P was true initially (antecedent) to a proof of the induction step $P \to [\alpha]P$ (premise). The induction rule is a derived rule, i.e., it is proved from other axioms and proof rules.

Lemma 7.2 (ind induction rule). *The loop induction rule ind is derived:*

$$\text{ind}\ \frac{P \vdash [\alpha]P}{P \vdash [\alpha^*]P}$$

Proof. Derived rule ind derives from axiom I using rule G for the inductive step:

$$\text{I}\ \dfrac{\land\text{R}\ \dfrac{\text{id}\ \dfrac{*}{P \vdash P} \qquad \text{G}\ \dfrac{\to\!\text{R}\ \dfrac{P \vdash [\alpha]P}{\vdash P \to [\alpha]P}}{P \vdash [\alpha^*](P \to [\alpha]P)}}{P \vdash P \land [\alpha^*](P \to [\alpha]P)}}{P \vdash [\alpha^*]P}$$

□

The induction rule ind derives easily from the induction axiom I. Its premise expresses that P is inductive, i.e., true after all runs of α if P was true before. If P is inductive (premise), then P is always true after any number of repetitions of α^* (succedent of conclusion) if P is true initially (antecedent of the conclusion).

Loop induction rule ind requires the postcondition P to occur verbatim in the succedent's antecedent. But the rule does not directly apply for sequents $\Gamma \vdash [\alpha^*]P$ in which the antecedent Γ merely implies P but does not literally include it. The difference is easily overcome with a use of the cut rule, though, which, with a cut of P, can get the required formula P into the antecedent for the ind rule:

$$\text{cut}\ \dfrac{\text{WR}\ \dfrac{\Gamma \vdash P, \Delta}{\Gamma \vdash P, [\alpha^*]P, \Delta} \qquad \text{WL,WR}\ \dfrac{\text{ind}\ \dfrac{P \vdash [\alpha]P}{P \vdash [\alpha^*]P}}{\Gamma, P \vdash [\alpha^*]P, \Delta}}{\Gamma \vdash [\alpha^*]P, \Delta}$$

Example 7.1. The only actual difference between the induction axiom I and the loop induction rule ind is that the latter already went a step further with the generalization rule G to discard the $[\alpha^*]$ modality, which makes rule ind more practical but also comes at a loss of precision. For example, the following simple dL formula is valid:

$$x \geq 0 \land v = 0 \to [(v := v+1; x' = v)^*]\, x \geq 0 \tag{7.4}$$

By induction axiom I, the valid formula (7.4) is equivalent to

$$x \geq 0 \land v = 0 \to x \geq 0 \land [(v := v+1; x' = v)^*]\big(x \geq 0 \to [v := v+1; x' = v]\, x \geq 0\big) \tag{7.5}$$

Nevertheless, the induction step to which rule ind would reduce the proof of (7.4) is not valid:

$$x \geq 0 \to [v := v+1; x' = v]\, x \geq 0 \tag{7.6}$$

The reason why, unlike the formula (7.5) resulting from the induction axiom I, the induction step (7.6) resulting from rule ind is *not* valid is simply that rule ind discards the modality $[(v:=v+1;x'=v)^*]$ by Gödel generalization G. Discarding this modality misses out on its effect on the state, which changes the values of x and v. But since the change of x in (7.6) depends on the value of v, the postcondition $x \geq 0$ of (7.4) cannot possibly suffice for the induction step (7.6).

The formula for the induction step needs to be *strengthened* to retain information about the value of v always remaining nonnegative when discarding the repetition modality when rule ind tries to prove (7.5) by generalization. That is why the next section investigates ways of using the formula $x \geq 0 \land v \geq 0$ as an invariant to prove (7.4) even if its postcondition did not talk about v.

7.3.3 Loop Invariants

Even if the induction rule ind captures the core essentials of induction, that rule does not necessarily result in a successful proof. Unlike the induction axiom I, which is an equivalence, the premise of the induction rule ind does not have to be valid even if its conclusion is valid, because the Gödel generalization rule G, which is used to derive rule ind from axiom I, discards modality $[\alpha^*]$. Where axiom I was overly precise with its induction step after repetition, rule ind loses all information about the loop in the induction step. It can happen that the formula $P \rightarrow [\alpha]P$ is not valid in all states, but only true after repeating α any number of times (which is what the subformula $[\alpha^*](P \rightarrow [\alpha]P)$ in axiom I expresses). In other words, the truth of $P \rightarrow [\alpha]P$ might depend on a certain property that happens to always hold after repeating α, but does not follow from assumption P alone. But to establish such an auxiliary property to always hold after repeating α would also need a proof by induction just like P.

In fact, that phenomenon is quite familiar from mathematics. Some inductive proofs require a stronger formulation of the induction hypothesis for the proof to succeed. The proof of Fermat's Last Theorem is not an inductive proof assuming that $a^n + b^n \neq c^n$ for $n > 2$ has already been proved for all smaller natural numbers.

Fortunately, the monotonicity rule M$[\cdot]$, which was discussed in Sect. 5.6.4, already provides a way of suitably generalizing the postcondition of $[\alpha^*]$ to another formula for which the premise of induction rule ind will be proved successfully. If P implies Q (premise of M$[\cdot]$), then $[\alpha]P$ implies $[\alpha]Q$ (conclusion).

Lemma 5.13 (M$[\cdot]$ monotonicity rule). *The* monotonicity rules *are sound:*

$$\text{M}[\cdot]\ \frac{P \rightarrow Q}{[\alpha]P \rightarrow [\alpha]Q} \qquad \text{M}\ \frac{P \rightarrow Q}{\langle\alpha\rangle P \rightarrow \langle\alpha\rangle Q}$$

The monotonicity rule M$[\cdot]$ turns the bare-bones induction rule ind into the more useful loop invariant rule, which proves a safety property P of a loop α^* by proving that some loop invariant J is true initially (first premise), is inductive (second premise), and finally implies the original postcondition P (third premise).

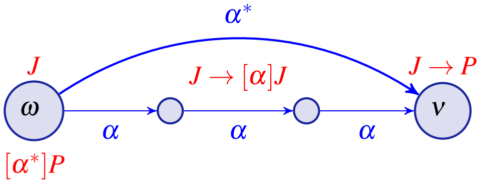
Lemma 7.3 (Loop invariant rule). *The loop invariant rule is derived:*

$$\text{loop}\ \dfrac{\Gamma \vdash J, \Delta \quad J \vdash [\alpha]J \quad J \vdash P}{\Gamma \vdash [\alpha^*]P, \Delta}$$

Proof. Rule loop is derived from the derived rule ind using a cut with $J \to [\alpha^*]J$ and weakening WL,WR (used without notice):

$$\text{cut}\,\dfrac{{\to}\text{R}\,\dfrac{\text{ind}\,\dfrac{J \vdash [\alpha]J}{J \vdash [\alpha^*]J}}{\Gamma \vdash J \to [\alpha^*]J, \Delta} \qquad {\to}\text{L}\,\dfrac{\Gamma \vdash J, \Delta \qquad \text{M}[\cdot]\,\dfrac{J \vdash P}{[\alpha^*]J \vdash [\alpha^*]P}}{\Gamma, J \to [\alpha^*]J \vdash [\alpha^*]P, \Delta}}{\Gamma \vdash [\alpha^*]P, \Delta}$$

□



First observe that the *inductive invariant* J occurs in all premises but not in the conclusion of rule loop. That means, whenever we apply the loop invariant rule to a desired conclusion, we get to choose what invariant J we want to use it for. Good choices of J will lead to a successful proof of the conclusion. Bad choices of J will stall the proof, because some of the premises cannot be proved.

The first premise of rule loop says that the initial state, about which we assume Γ (and that Δ does not hold), satisfies the invariant J, i.e., the invariant is initially true. The second premise of rule loop shows that the invariant J is *inductive*. That is, whenever J was true before running the loop body α, then J is always true again after running α. The third premise of rule loop shows that the invariant J is strong enough to imply the postcondition P that the conclusion was interested in.

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

Taking a step back, these three premises correspond to the proof steps one would use to show that the contract of an ordinary program with a **requires**() contract Γ

(and not Δ), an **ensures**(P) contract, and a loop invariant J is correct. Now, we have this reasoning in a more general and formally more precisely defined context. We no longer need to appeal to intuition to justify why such a proof rule is fine, but can evoke a soundness proof for loop. We will also no longer be limited to informal arguments to justify invariance for a program but can do actual solid and rigorous formal proofs if we combine proof rule loop with the other proof rules from Chap. 6.

Invariants are crucial concepts for conventional programs and continue to be even more crucial for cyber-physical systems, where change is ubiquitous and any identification of aspects that remain unchanged over time is a blessing.

Of course, the search for suitable loop invariants J to be used with the loop invariant rule can be as much of a challenge as the search for invariants in mathematics. Yet, the fact that the difference between the equivalence in the induction axiom I and the induction steps of rules loop and ind is the absence of the $[\alpha^*]$ modality provides some guidance on what kind of information loop invariants J need. Loop invariants J may need to communicate something else that is also always true after running α^* and carries just information about the past behavior during α^* to imply that they are preserved after running α once more.

Example 7.2 (Stronger invariants). Consider an obvious example of a purely discrete loop to illustrate the rôle of loop invariants in proving the safety of loops:

$$x \geq 8 \land 5 \geq y \land y \geq 0 \rightarrow [(x:=x+y; y:=x-2\cdot y)^*]x \geq 0$$

This formula is valid. A proof with loop invariant J starts like this:

$$\dfrac{\dfrac{x \geq 8 \land 5 \geq y \land y \geq 0 \vdash J \quad J \vdash [x:=x+y; y:=x-2\cdot y]J \quad J \vdash x \geq 0}{x \geq 8 \land 5 \geq y \land y \geq 0 \vdash [(x:=x+y; y:=x-2\cdot y)^*]x \geq 0}\text{loop}}{\vdash x \geq 8 \land 5 \geq y \land y \geq 0 \rightarrow [(x:=x+y; y:=x-2\cdot y)^*]x \geq 0}\rightarrow\text{R}$$

A direct proof with the postcondition $x \geq 0$ as invariant J cannot succeed, because the induction step is not valid, since $x \geq 0$ is not guaranteed to be true after $x:=x+y$ if the inductive hypothesis only guarantees $x \geq 0$ about the previous state if y might be negative. The loop invariant J needs to imply the postcondition $x \geq 0$ but also contain additional information about the variable y that the change of x depends on.

The initial condition $x \geq 8 \land 5 \geq y \land y \geq 0$ also fails to be an invariant J since its induction step is not valid, because $5 \geq y$ is no longer guaranteed to be true after $x:=x+y; y:=x-2\cdot y$, e.g., if $x=8, y=0$ holds initially. The loop invariant J needs to be implied by the precondition, but may have to be weaker because the precondition itself does not have to remain true always when repeating the loop.

The loop invariant J, thus, has to be somewhere between the precondition (first premise) and the postcondition (third premise). It needs to involve bounds on both x and y, because the change of x depends on y and vice versa (second premise). The first assignment $x:=x+y$ obviously preserves $x \geq 0$ if also $y \geq 0$. The loop body obviously preserves this $y \geq 0$ if $x \geq y$. Indeed, the conjunction $x \geq y \land y \geq 0$ succeeds as loop invariant J:

$$
{\to}\mathrm{R}\ \dfrac{\mathrm{loop}\ \dfrac{\mathbb{R}\dfrac{*}{x{\ge}8\land 5{\ge}y\land y{\ge}0\vdash J}\quad [;]\dfrac{[:=]\dfrac{\mathbb{R}\dfrac{*}{J\vdash x+y\ge x-y\land x-y\ge 0}}{J\vdash [x:=x+y][y:=x-2\cdot y]J}}{J\vdash [x:=x+y;\, y:=x-2\cdot y]J}\quad \mathbb{R}\dfrac{*}{J\vdash x{\ge}0}}{x\ge 8\land 5\ge y\land y\ge 0\vdash [(x:=x+y;\, y:=x-2\cdot y)^*]x\ge 0}}{\vdash x\ge 8\land 5\ge y\land y\ge 0\to [(x:=x+y;\, y:=x-2\cdot y)^*]x\ge 0}
$$

A similar proof uses the loop invariant $x\geq 0\land y\geq 0$ to prove Example 7.1.

Note 38 (Of loop invariants and relay races) Loop invariants J are the proof analogue of a relay race. The initial state needs to show they have the baton J. Every state along the way after repeating α^* any number of times needs to wait to receive the baton J and then pass the baton J to the next state after running the next leg α of the relay race α^*. When the final state receives the baton J, that baton needs to carry enough information to meet the goal's safety condition P. Finding a loop invariant J is like designing the baton that makes all these passing phases work out as easily as possible.

7.3.4 Contextual Soundness Requirements

Since the loop rule derives via monotonicity rule M[·] from rule ind, which derives via Gödel's generalization G, it should not come as a surprise that it is crucial for soundness that the sequent context formulas Γ and Δ disappear from the middle and last premises of loop. It is equally soundness-critical that no context Γ,Δ carries over to the premise of rules G,M[·],M,ind. All those premises result from discarding the $[\alpha^*]$ modality, which ignores its effect. That is sound as long as no context Γ,Δ is preserved, which represents assumptions about the initial state before $[\alpha^*]$, which may no longer be true after $[\alpha^*]$. For the loop rule, information Γ,Δ about the initial state is only available to show that J is initially true (first premise), but no longer during the induction step (second premise) or use case (third premise).

Example 7.3 (No context). The context Γ,Δ cannot be kept in the ind rule without losing soundness:

$$
\text{↯}\ \dfrac{x=0, x\leq 1\vdash [x:=x+1]x\leq 1}{x=0, x\leq 1\vdash [(x:=x+1)^*]x\leq 1}
$$

This inference is unsound, because the premise is valid but the conclusion is not, since $x\leq 1$ will be violated after two repetitions. Even if $x=0$ is assumed initially (antecedent of conclusion), it cannot be assumed in the induction step (premise), because it is no longer true after iterating the loop any nonzero number of times. Almost the same counterexample shows that the middle premise of the loop rule cannot keep a context soundly. The following counterexample shows that the third premise of rule loop also cannot keep a context without losing soundness:

$$\text{↯}\,\frac{x=0\vdash x\geq 0 \quad x\geq 0\vdash [x:=x+1]x\geq 0 \quad x=0, x\geq 0\vdash x=0}{x=0\vdash [(x:=x+1)^*]x=0}$$

With some more thought, assumptions about *constant* parameters that cannot change during the HP α^* could be kept around without endangering soundness. This can be proved with the help of the vacuity axiom V from Sect. 5.6.2 (Exercise 7.8).

With Lemma 7.3, the loop invariant rule already has a simple and elegant soundness proof that simply derives it by monotonicity M[·] (Lemma 5.13) from the induction rule ind, which, in turn, is derived using Gödel's generalization rule G from the induction axiom I. Since loop invariants are such a fundamental concept, and since Example 7.3 just made us painfully aware how careful we need to be to keep CPS reasoning principles sound, we provide a second soundness proof directly from the semantics even if that proof is entirely redundant and more complicated than the first proof of Lemma 7.3.

Proof (of Lemma 7.3). In order to prove that rule loop is sound, we assume that all its premises are valid and need to show that its conclusion is valid, too. So let $\vDash \Gamma\vdash J,\Delta$ and $\vDash J\vdash[\alpha]J$ and $\vDash J\vdash P$. In order to prove that $\vDash\Gamma\vdash[\alpha^*]P,\Delta$, consider any state ω and show that $\omega\in[\![\Gamma\vdash[\alpha^*]P,\Delta]\!]$. If one of the formulas $Q\in\Gamma$ does not hold in ω (that is $\omega\notin[\![Q]\!]$) or if one of the formulas in $Q\in\Delta$ holds in ω ($\omega\in[\![Q]\!]$), then there is nothing to show, because the formula that the sequent $\Gamma\vdash[\alpha^*]P,\Delta$ represents already holds in ω, either because one of the conjunctive assumptions Γ is not met in ω or because one of the other disjunctive succedents Δ already holds. Consequently, let all $Q\in\Gamma$ be true in ω and all $Q\in\Delta$ be false in ω or else there is nothing to show.

In that case, however, the first premise implies that $\omega\in[\![J]\!]$ because all its assumptions (which are the same Γ) are met in ω and all alternative succedents (which are the same Δ) do not already hold.[2]

In order to show that $\omega\in[\![[\alpha^*]P]\!]$, consider any run $(\omega,\nu)\in[\![\alpha^*]\!]$ from the initial state ω to some state ν and show that $\nu\in[\![\alpha]\!]$. According to the semantics of loops from Chap. 3, $(\omega,\nu)\in[\![\alpha^*]\!]$ if and only if, for some natural number $n\in\mathbb{N}$ that represents the number of loop iterations, there is a sequence of states $\mu_0,\mu_1,\dots,\mu_n$ such that $\mu_0=\omega$ and $\mu_n=\nu$ such that $(\mu_i,\mu_{i+1})\in[\![\alpha]\!]$ for all $i<n$. The proof that $\mu_n\in[\![J]\!]$ is now by induction on n.

0. If $n=0$, then $\nu=\mu_0=\mu_n=\omega$, which implies by the first premise that $\nu\in[\![J]\!]$.
1. By induction hypothesis, $\mu_n\in[\![J]\!]$. By the second premise, $\vDash J\vdash[\alpha]J$, in particular for state μ_n we have $\mu_n\in[\![J\rightarrow[\alpha]J]\!]$, recalling the semantics of sequents. Combined with the induction hypothesis, this implies $\mu_n\in[\![[\alpha]J]\!]$, which means that $\mu\in[\![J]\!]$ for all states μ such that $(\mu_n,\mu)\in[\![\alpha]\!]$. Hence, $\mu_{n+1}\in[\![J]\!]$ because $(\mu_n,\mu_{n+1})\in[\![\alpha]\!]$.

This implies, in particular, that $\nu\in[\![J]\!]$, because $\mu_n=\nu$. By the third premise, $\vDash J\vdash P$. In particular, $\nu\in[\![J\rightarrow P]\!]$, which with $\nu\in[\![J]\!]$ implies $\nu\in[\![P]\!]$. This con-

[2] In future soundness proofs, we will fast-forward to this situation right away, but it is instructive to see the full argument once.

cludes the soundness proof, since ν was an arbitrary state such that $(\omega,\nu) \in [\![\alpha^*]\!]$, so $\omega \in [\![[\alpha^*]P]\!]$. □

7.4 A Proof of a Happily Repetitive Bouncing Ball

Now that he understands the principles of how to prove loops in CPSs, Quantum is eager to put these skills to use. Quantum wants to relieve himself of his acrophobic fears once and for all by proving that he won't ever have to be afraid of excess heights $> H$ again nor of falling through the cracks in the ground to heights < 0.

Abbreviations have served Quantum well in trying to keep proofs on one page:

$$\begin{aligned} A &\overset{\text{def}}{\equiv} 0 \le x \land x = H \land v = 0 \land g > 0 \land 1 \ge c \ge 0 \\ B(x,v) &\overset{\text{def}}{\equiv} 0 \le x \land x \le H \\ x''.. &\overset{\text{def}}{\equiv} \{x' = v, v' = -g\} \end{aligned}$$

Note the somewhat odd abbreviation for the differential equation just to condense notation. With these abbreviations, the bouncing-ball conjecture (7.1) turns into

$$A \to [(x''..;(?x=0;v:=-cv \cup ?x \ne 0))^*]B(x,v) \tag{7.1*}$$

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

$$\to\text{R}\,\frac{A \vdash [(x''..;(?x=0;v:=-cv \cup ?x \ne 0))^*]B(x,v)}{\vdash A \to [(x''..;(?x=0;v:=-cv \cup ?x \ne 0))^*]B(x,v)}$$

Its premise leaves a loop to worry about, which gives Quantum a chance to practice what he learned in this chapter.

The first thing that Quantum will need for the proof of (7.1) is the appropriate choice for the invariant J to be used in the loop invariant proof rule loop. Quantum will use a dL formula $\mathrm{j}(x,v)$ for the invariant when instantiating J in the proof rule loop. But Quantum is still a little unsure about how exactly to define that formula $\mathrm{j}(x,v)$, not an unusual situation when trying to master the understanding of a CPS. Can you think of a good choice for the formula $\mathrm{j}(x,v)$ to help Quantum?

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

I don't know about you, but Quantum settles for the choice of using the postcondition as an invariant, because that is what he wants to show about the behavior:

$$\mathrm{j}(x,v) \overset{\text{def}}{\equiv} 0 \le x \land x \le H \tag{7.7}$$

Because Quantum is so proud of his wonderful invariant $\mathrm{j}(x,v)$, he even uses it to perform a generalization with the newly acquired skill of the generalization proof rule MR in the inductive step to completely separate the proof about the differential equation and the proof about the bouncing dynamics.[3] Quantum conducts the proof in Fig. 7.2.

$$
{\scriptstyle\to\mathrm{R}}\dfrac{
{\scriptstyle\text{loop}}\dfrac{
A\vdash \mathrm{j}(x,v)
\quad
{\scriptstyle[;]}\dfrac{
{\scriptstyle\text{MR}}\dfrac{
\mathrm{j}(x,v)\vdash[x''..]\mathrm{j}(x,v)
\quad
{\scriptstyle[\cup]}\dfrac{
{\scriptstyle\land\mathrm{R}}\dfrac{
{\scriptstyle[;]}\dfrac{
{\scriptstyle[?]}\dfrac{
{\scriptstyle[:=]}\dfrac{\mathrm{j}(x,v),x=0\vdash \mathrm{j}(x,-cv)}{\mathrm{j}(x,v),x=0\vdash[v:=-cv]\mathrm{j}(x,v)}
}{\mathrm{j}(x,v)\vdash[?x=0][v:=-cv]\mathrm{j}(x,v)}
}{\mathrm{j}(x,v)\vdash[?x=0;v:=-cv]\mathrm{j}(x,v)}
\quad
{\scriptstyle[?]}\dfrac{\mathrm{j}(x,v),x\neq0\vdash\mathrm{j}(x,v)}{\mathrm{j}(x,v)\vdash[?x\neq0]\mathrm{j}(x,v)}
}{\mathrm{j}(x,v)\vdash[?x=0;v:=-cv]\mathrm{j}(x,v)\land[?x\neq0]\mathrm{j}(x,v)}
}{\mathrm{j}(x,v)\vdash[?x=0;v:=-cv\cup ?x\neq0]\mathrm{j}(x,v)}
}{\mathrm{j}(x,v)\vdash[x''..][?x=0;v:=-cv\cup ?x\neq0]\mathrm{j}(x,v)}
}{\mathrm{j}(x,v)\vdash[x''..;(?x=0;v:=-cv\cup ?x\neq0)]\mathrm{j}(x,v)}
\quad
\mathrm{j}(x,v)\vdash B(x,v)
}{A\vdash\big[\big(x''..;(?x=0;v:=-cv\cup ?x\neq0)\big)^*\big]B(x,v)}
}{\vdash A\to\big[\big(x''..;(?x=0;v:=-cv\cup ?x\neq0)\big)^*\big]B(x,v)}
$$

Fig. 7.2 Sequent calculus proof shape for bouncing ball (7.1)

The proof in Fig. 7.2 has five premises remaining to be proved. Quantum is pretty sure how to prove the first premise ($A\vdash \mathrm{j}(x,v)$), corresponding to the initial condition, because $0\le x\le H$ is true initially as $0\le x=H$ follows from A. Quantum also knows how to prove the last premise ($\mathrm{j}(x,v)\vdash B(x,v)$), because the invariant $\mathrm{j}(x,v)$ from (7.7) is equal to the desired postcondition $B(x,v)$, so this is proved by the identity rule id.

But Quantum runs into unforeseen(?) trouble with the inductive step in the middle. While the third and fourth premise succeed, the second premise $\mathrm{j}(x,v)\vdash[x''..]\mathrm{j}(x,v)$ with the differential equation resists all proof attempts for the choice (7.7). That makes sense, because, even if the current height is bounded by $0\le x\le H$ before the differential equation, there is no reason to believe it will remain bounded afterwards if this is all we know about the bouncing ball. If the ball were just below $x=H$, it would still ultimately exceed H if its velocity were too big.

Ah, right! We actually found that out about the bouncing ball in Chap. 4 already when we were wondering under what circumstances it might be safe to let a ball bounce around. As a matter of fact, everything we learned by the Principle of Cartesian Doubt about when it is safe to start a CPS is valuable information to preserve in the invariant. If it wasn't safe to start a CPS in a state, chances are, it wouldn't be safe either if we kept it running in such a state as we do in an inductive step.

Well, so Quantum found a (poor) choice of an invariant $\mathrm{j}(x,v)$ in (7.7) that just cannot be proved because of the inductive step. What to do?, wonders Quantum.

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

[3] This is not necessary and Quantum might just as well not have used MR and gone for a direct proof using $[']$ right away instead. But it does save us some space on the page, and also showcases a practical use of proof rule MR.

There was trouble in the induction step, because $x \leq H$ could not be proved to be inductive. But Quantum does not despair. Quantum can demand a little less from the invariant and use the following weaker choice for $\mathrm{j}(x,v)$ instead of (7.7):

$$\mathrm{j}(x,v) \overset{\text{def}}{\equiv} x \geq 0 \tag{7.8}$$

Armed with this new choice for an invariant, Quantum quickly gets to work constructing a new proof for (7.1). After frantically scribbling a couple of pages with sequent proofs, Quantum experiences a *déjà vu* and notices that his new proof has exactly the same form as the last sequent proof he began, just with a different choice for the logical formula $\mathrm{j}(x,v)$ to be used as the invariant when applying the loop rule with the choice (7.8) rather than (7.7) for $\mathrm{j}(x,v)$. Fortunately, Quantum already worked with an abbreviation last time he started a proof, so it is actually not surprising after all to see that the proof structure stays exactly the same and that the particular choice of $\mathrm{j}(x,v)$ only affects the premises, not the way the proof unraveled its program statements in the modalities.

Inspecting the five premises of the above sequent proof attempt in light of the improved choice (7.8) for the invariant, Quantum is delighted to find that the inductive step works out just fine. The height stays above ground always by construction with the evolution domain constraint $x \geq 0$ and is not changed in the subsequent discrete bouncing control. The initial condition ($A \vdash \mathrm{j}(x,v)$) also works out alright, because $0 \leq x$ was among the assumptions in A. Only this time, the last premise ($\mathrm{j}(x,v) \vdash B(x,v)$) falls apart, because $x \geq 0$ is not at all enough to conclude the part $x \leq H$ of the postcondition. What's a ball to do to get himself verified these days?

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

Quantum takes the lesson from Cartesian Doubt to heart and realizes that the invariant needs to transport enough information about the state of the system to make sure the inductive step has a chance of holding true. In particular, the invariant desperately needs to preserve knowledge about the velocity, because how the height changes depends on the velocity (after all the differential equation reads $x' = v, \ldots$), so it would be hard to get a handle on height x without first understanding how velocity v changes, which it does in $v' = -g$ and at the bounce. Indeed, this is an entirely syntactic reason why neither (7.7) nor (7.8) could have worked out as invariants for the proof of (7.1). They only mention the height x, but how the height changes in the bouncing-ball HP depends on the velocity, which also changes. So unless the invariant preserves knowledge about v, it cannot possibly guarantee much about height x, except the fact $x \geq 0$ from the evolution domain constraint, which does not suffice to prove the postcondition $0 \leq x \leq H$.

Fine, so Quantum quickly discards the failed invariant choice from (7.7), which he is no longer quite so proud of, and also gives up on the weaker version (7.8), but instead shoots for a stronger invariant, which is surely inductive and strong enough to imply safety:

$$\mathrm{j}(x,v) \overset{\text{def}}{\equiv} x = 0 \land v = 0 \tag{7.9}$$

This time, Quantum has learned his lesson and won't blindly set out to prove the property (7.1) from scratch again, but, rather, be clever about it and realize that he is still going to find the same shape of the sequent proof attempt above, just with, once again, a different choice for the invariant $\mathrm{j}(x,v)$. So Quantum quickly jumps to conclusions and inspects the famous 5 premises of the above sequent proof attempt. This time, the postcondition is a piece of cake and the inductive step works like a charm (no velocity, no height, no motion). But the initial condition is giving Quantum quite a bit of a headache, because there is no reason to believe the ball would initially lie flat on the ground with velocity zero.

For a moment there, Quantum fancied the option of simply editing the initial condition A to include $x = 0$, because that would make this proof attempt work out just fine. But then he realized that this would mean that he would from now on be doomed to always start the day at speed zero on the ground, which would not lead to all that much excitement for a cheerful bouncing ball. That option would be safe, but a bit too much so for lack of motion.

What, then, is poor Quantum supposed to do to finally get a proof without losing all those exciting initial conditions?

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

This time, Quantum thinks about the invariant question really hard and has a smart idea. Thinking back to where the idea of the loop invariants came from in the first place, they are replacements for the postcondition P that make $[\alpha^*](P \to [\alpha]P)$ provable despite discarding the $[\alpha^*]$ modality. They are stronger versions of the postcondition P, and need to at least imply that the postcondition P always holds after running α once, but, in fact, even need to imply they themselves continue to hold after α. For this to work out, their rôle is to capture whatever we still need to know about the previous runs of α^*.

If the loop invariant has to work for any number of loop iterations, it certainly has to work for the first few loop iterations. In particular, the loop invariant J is not unlike an intermediate condition of $\alpha;\alpha$. Quantum already identified an intermediate condition for the single-hop bouncing ball in Sect. 4.8.1. Maybe that will prove useful as an invariant, too:

$$\mathrm{j}(x,v) \overset{\text{def}}{\equiv} 2gx = 2gH - v^2 \land x \geq 0 \tag{7.10}$$

After all, an invariant is something like a permanent intermediate condition, i.e., an intermediate condition that keeps on working out alright for all future iterations. The bouncing ball is not yet sure whether this will work but it seems worth trying!

The shape of the proof in Fig. 7.2 again stays exactly the same, just with a different choice of $\mathrm{j}(x,v)$, this time coming from (7.10). The remaining famous five premises are then proved easily. The first premise $A \vdash \mathrm{j}(x,v)$ is proved using $x = H$ and $v = 0$:

$$0 \leq x \land x = H \land v = 0 \land g > 0 \land 1 \geq c \geq 0 \vdash 2gx = 2gH - v^2 \land x \geq 0$$

Expanding the abbreviations, the second premise $\mathrm{j}(x,v) \vdash [x''..]\mathrm{j}(x,v)$ is

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

a proof that we have seen in previous chapters (Exercise 7.1). The third premise $\mathrm{j}(x,v), x = 0 \vdash \mathrm{j}(x,-cv)$ is

$$2gx = 2gH - v^2 \wedge x \geq 0, x = 0 \vdash 2gx = 2gH - (-cv)^2 \wedge x \geq 0$$

which would be proved 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. But we could prove this third premise easily if we edited the definition of the initial condition A to include $c = 1$. That is not the most general statement about bouncing balls, but let's happily settle for it till Exercise 7.5. Even then, however, we still need to augment $\mathrm{j}(x,v)$ to include $c = 1$ as well, since we otherwise would have lost this knowledge before we need it in the third premise. Having misplaced critical pieces of knowledge is a phenomenon you may encounter when you are conducting proofs. In such cases, you should trace where you lost the assumption in the first place and put it back in. But then you have also learned something valuable about your system, namely which assumptions are crucial for the correct functioning of which part of the system.

The fourth premise, $\mathrm{j}(x,v), x \geq 0 \vdash \mathrm{j}(x,v)$ is proved splendidly whatever the abbreviations stand for simply using the identity rule id. In fact, Quantum could have noticed this earlier already but might have been distracted by his search for a good choice for the invariant $\mathrm{j}(x,v)$. This is but one indication of the fact that it may pay to take a step back from a proving effort and critically reflect on what all the pieces of the argument rely on exactly. Finally, the fifth premise $\mathrm{j}(x,v) \vdash B(x,v)$, which is

$$2gx = 2gH - v^2 \wedge x \geq 0 \vdash 0 \leq x \wedge x \leq H$$

is proved by arithmetic as long as we know $g > 0$. This condition is already included in A. But we still managed to forget about that in our invariant $\mathrm{j}(x,v)$. So, again, the constant parameter assumption $g > 0$ should have been included in the invariant $\mathrm{j}(x,v)$, which, overall, should have been defined as

$$\mathrm{j}(x,v) \overset{\text{def}}{\equiv} 2gx = 2gH - v^2 \wedge x \geq 0 \wedge (c = 1 \wedge g > 0) \tag{7.11}$$

This is nearly the same definition as (7.10) except that assumptions about the system parameter choices are carried through. The last two conjuncts are trivially invariant, because neither c nor g changes while the little bouncing ball falls. As written, the loop invariant rule, unfortunately, still needs to have these constant assumptions included in the invariant, because it wipes the entire context Γ, Δ, which is crucial for soundness (Sect. 7.3.4). Exercise 7.8 investigates simplifications for this nuisance that will enable you to elide the trivial constant part $c = 1 \wedge g > 0$ from the invariant. Redoing the proof with the new loop invariant (7.11) will succeed, as will the proof in Sect. 7.5.

For the record, we now really have a full sequent proof of the undamped bouncing ball with repetitions. Quantum is certainly quite thrilled about this achievement!

Proposition 7.1 (Quantum is safe). *This* dL *formula has a proof and is, thus, valid:*

$$
\begin{aligned}
&0 \le x \land x = H \land v = 0 \land g > 0 \land 1 = c \rightarrow \\
&[\big(\{x' = v, v' = -g \,\&\, x \ge 0\}; (?x = 0; v := -cv \cup ?x \ne 0)\big)^*](0 \le x \land x \le H)
\end{aligned}
\tag{7.12}
$$

Since invariants are a crucial part of a CPS design, you are encouraged to describe invariants in your hybrid programs. KeYmaera X will make use of the invariants annotated using the @invariant contract in hybrid programs to simplify your proof effort. But KeYmaera X solved Exercise 7.8 already, so it does not require a list of the constant expressions in the @invariant contracts. It is a good idea to rephrase (7.12) by explicitly including the invariant contract in the hybrid program for documentation as well as verification purposes:

$$
\begin{aligned}
&0 \le x \land x = H \land v = 0 \land g > 0 \land 1 = c \rightarrow \\
&\quad [\big(\{x' = v, v' = -g \,\&\, x \ge 0\}; \\
&\quad\;\; (?x = 0; v := -cv \cup ?x \ne 0)\big)^* \text{@invariant}(2gx = 2gH - v^2 \land x \ge 0)] \\
&\quad (0 \le x \land x \le H)
\end{aligned}
\tag{7.13}
$$

Indeed, assumptions about constant parameters, which are trivially invariant, do not need to be listed, as the next section will explain.

7.5 Splitting Postconditions into Separate Cases

The invariant $\mathrm{j}(x,v)$ from formula (7.10) was not quite enough for proving the bouncing-ball property (7.12) since we need *constant parameter* assumptions from the modified invariant (7.11). Redoing the proof with the new invariant succeeds. But it would be easier if there was a way of reusing the old proof by threading the misplaced assumptions through to where we need them. Of course, we cannot simply add assumptions into the middle of a proof without losing soundness (Sect. 7.3.4). But Quantum wonders whether we might get away with doing that if it is merely a matter of adding assumptions about constant parameters such as $c = 1 \land g > 0$?

Indeed, there are two interesting insights about clever proof structuring that we can learn from this desire. One insight is an efficient way of proving the preservation of assumptions about constant parameters. The other is about modularly separating the reasoning into proofs for separate postconditions.

The dynamic axioms from Chap. 5 and the sequent proof rules from Chap. 6 decompose correctness analysis along the top-level operators, which, e.g., split the analysis into separate questions along the top-level operators in the hybrid programs. But it is also possible to split the reasoning along the postcondition to show

$[\alpha](P \land Q)$ by proving $[\alpha]P$ and $[\alpha]Q$ separately. If the HP α satisfies both the safety postcondition P and the safety postcondition Q, then it also satisfies the safety postcondition $P \land Q$, and vice versa, using the following result from Sect. 5.6.1.

Lemma 5.10 ($[]\land$ boxes distribute over conjunctions). *This axiom is sound:*

$$[]\land \quad [\alpha](P \land Q) \leftrightarrow [\alpha]P \land [\alpha]Q$$

The axiom $[]\land$ can decompose the box modality in the induction step along the conjunction in the loop invariant $\mathrm{j}(x,v) \land q$ to conduct separate proofs that $\mathrm{j}(x,v)$ is inductive (second premise in Fig. 7.3) and that the additional invariant q, which is defined as $c = 1 \land g > 0$, is inductive (third premise). In Fig. 7.3, bb denotes the loop body of the bouncing ball (7.12). Observe how the induction proof in the second premise is literally the same proof as the previous proof in Fig. 7.2, except that the missing assumption q is now available. The remaining proof that the additional loop invariant q is also inductive is isolated in the third premise.

$$\text{loop}\,\dfrac{\mathbb{R}\,\dfrac{*}{A \vdash \mathrm{j}(x,v) \land q} \qquad []\land\,\dfrac{\land L\,\dfrac{\land R\,\dfrac{\dfrac{\text{see Fig. 7.2}}{\mathrm{j}(x,v), q \vdash [\mathrm{bb}]\mathrm{j}(x,v)} \qquad V\,\dfrac{*}{\mathrm{j}(x,v), q \vdash [\mathrm{bb}]q}}{\mathrm{j}(x,v), q \vdash [\mathrm{bb}]\mathrm{j}(x,v) \land [\mathrm{bb}]q}}{\mathrm{j}(x,v) \land q \vdash [\mathrm{bb}]\mathrm{j}(x,v) \land [\mathrm{bb}]q}}{\mathrm{j}(x,v) \land q \vdash [\mathrm{bb}](\mathrm{j}(x,v) \land q)} \qquad \mathbb{R}\,\dfrac{*}{\mathrm{j}(x,v) \land q \vdash B(x,v)}}{A \vdash [\mathrm{bb}^*]B(x,v)}$$

Fig. 7.3 Sequent calculus proof for bouncing ball (7.12) with split

There is an additional interesting twist in the proof in Fig. 7.3, though. The proof of its third premise establishes that formula q is inductive for the bouncing ball bb. This could be proved by successively decomposing the HP bb using the various dynamic axioms for sequential compositions, nondeterministic choices, differential equations, etc. While that proof would work, it is significantly more efficient to prove it in a single step using the axiom V (from Sect. 5.6.2) for postconditions of box modalities that do not change any of the variables in the postcondition, so that the postcondition is true after all runs of the HP if only it is true before.

Lemma 5.11 (V vacuous axiom). *The* vacuous axiom *is sound:*

$$\mathrm{V} \quad p \to [\alpha]p \quad (FV(p) \cap BV(\alpha) = \emptyset)$$

where no free variable of p *is bound (written) in* α.

When used like this, the axioms $[]\land$ and V justify that constant parameter assumptions can be kept around without any harm to the proof (Exercise 7.8).

7.6 Summary

This chapter focused on developing and using the concept of invariants for CPS. Invariants enable us to prove properties of CPSs with loops, a problem of ubiquitous significance, because hardly any CPS get by without repeating some operations in a control loop. *Invariants constitute the single most insightful and most important piece of information about a CPS, because they tell us what we can rely on no matter how long a CPS runs.* Invariants are a fundamental force of computer science, and are just as important in mathematics and physics.

The axioms and proof rules investigated in this chapter are summarized in Fig. 7.4. While the loop invariant rule (loop) is the most practical approach for loops, the induction axiom I is an equivalence and explains the core principle of loop induction more directly. The loop invariant rule loop also derives directly from the induction axiom I by monotonicity rule M[·] and generalization rule G.

$$\text{I}\quad [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \to [\alpha]P)$$

$$\text{G}\quad \frac{P}{[\alpha]P}$$

$$\text{M}[\cdot]\quad \frac{P \to Q}{[\alpha]P \to [\alpha]Q}$$

$$\text{loop}\quad \frac{\Gamma \vdash J, \Delta \quad J \vdash [\alpha]J \quad J \vdash P}{\Gamma \vdash [\alpha^*]P, \Delta}$$

$$\text{MR}\quad \frac{\Gamma \vdash [\alpha]Q, \Delta \quad Q \vdash P}{\Gamma \vdash [\alpha]P, \Delta}$$

$$[]\land\quad [\alpha](P \land Q) \leftrightarrow [\alpha]P \land [\alpha]Q$$

$$\text{V}\quad p \to [\alpha]p \quad (FV(p) \cap BV(\alpha) = \emptyset)$$

Fig. 7.4 Summary of proof rules for loops, generalization, monotonicity, and splitting boxes

The development that led to invariants has some interesting further consequences especially for finding bugs in CPSs by unrolling loops and disproving the resulting premises. But this bounded-model-checking principle is of limited use for ultimately verifying safety, because it only considers the system some finite number of steps in the future. This chapter focused on proving $[\alpha^*]P$ formulas, which were based on *invariants*, so properties that do not change. The discussion of proof techniques for proving $\langle\alpha^*\rangle P$ formulas will be postponed till Sect. 17.4, which will use *variants*, so properties that do change and steadily make progress toward the goal P.

In our effort to help the bouncing ball Quantum succeed with his proof, we saw a range of reasons why an inductive proof may not work out and what needs to be done to adapt the invariant.

7.7 Appendix

This appendix provides an alternative way of motivating the loop induction rule only from successively unwinding a loop with the iteration axiom $[^*]$ that works without using the more elegant induction axiom I.

7.7.1 Loops of Proofs

The iteration axiom $[^*]$ can be used to turn a safety property of a loop

$$A \rightarrow [\alpha^*]B \tag{7.14}$$

into the following equivalent dL formula:

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

What can we do to prove that loop? Investigating our proof rules from previous chapters, there is exactly one that addresses loops: the iteration $[^*]$ axiom again. Recall that, unlike sequent proof rules, axioms do not dictate 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

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

Let's do that again because that was so much fun and use the $[^*]$ axiom on the only occurrence of $[\alpha^*]B$ to obtain

$$A \rightarrow B \land [\alpha](B \land [\alpha](B \land [\alpha][\alpha^*]B)) \tag{7.15}$$

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 axiom $[^*]$ so far, notice that (7.15) still allows us to learn something about α and whether it always satisfies B when repeating α. Since $[^*]$ is an equivalence axiom, formula (7.15) still expresses the same thing as (7.14), i.e., that postcondition B always holds after repeating α when A was true in the beginning. Yet, (7.15) explicitly singles out the first three runs of α. Let's make this more apparent with the derived axiom $[]\land$ for box splitting from Sect. 5.6.1. Using this valid equivalence turns (7.15) into

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

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

$$A \to B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha][\alpha^*]B \tag{7.16}$$

$$
\begin{array}{rl}
 & A \vdash B \qquad A \vdash [\alpha]B \quad A \vdash [\alpha][\alpha]B \quad A \vdash [\alpha][\alpha][\alpha][\alpha^*]B \\ \hline
\land R,\land R,\land R & \quad A \vdash B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha][\alpha^*]B \\ \hline
[]\land & \quad A \vdash B \land [\alpha]B \land [\alpha]([\alpha]B \land [\alpha][\alpha][\alpha^*]B) \\ \hline
[]\land & \quad A \vdash B \land [\alpha]B \land [\alpha][\alpha](B \land [\alpha][\alpha^*]B) \\ \hline
[]\land & \quad A \vdash B \land [\alpha]\big(B \land [\alpha](B \land [\alpha][\alpha^*]B)\big) \\ \hline
[*] & \quad A \vdash B \land [\alpha](B \land [\alpha][\alpha^*]B) \\ \hline
[*] & \quad A \vdash B \land [\alpha][\alpha^*]B \\ \hline
[*] & \quad A \vdash [\alpha^*]B
\end{array}
$$

Fig. 7.5 Loops of proofs: iterating and splitting the box

Fig. 7.5 illustrates the proof construction so far.[4] Looking at it this way, (7.16) could be more useful than the original (7.14), because, even though the two formulas are equivalent, (7.16) 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:

$$A \to B \land [\alpha]B \land [\alpha][\alpha]B \tag{7.17}$$

If this formula is not valid, then, certainly, neither is (7.16) and, thus, neither is the original (7.14). Hence, if we find a counterexample to (7.17), we disproved (7.16) and (7.14). That can actually be rather useful!

However, if (7.17) is valid, we do not know whether (7.16) and (7.14) 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 (7.15) to obtain

$$A \to B \land [\alpha](B \land [\alpha](B \land [\alpha](B \land [\alpha][\alpha^*]B))) \tag{7.18}$$

Or, equivalently, use axiom $[*]$ on (7.16) to obtain the equivalent:

$$A \to B \land [\alpha]B \land [\alpha][\alpha]B \land [\alpha][\alpha][\alpha](B \land [\alpha][\alpha^*]B) \tag{7.19}$$

[4] Observe the $\land$R,$\land$R,$\land$R at the top, which is not to be taken as an indication that the proof is stuttering, but merely meant as a notational reminder that the $\land$R proof rule was actually used three times for that step. Because it will frequently simplify the notation, we will take the liberty of applying multiple rules at once like that without saying which derivation it was exactly. In fact, mentioning $\land$R three times seems a bit repetitive, so we simply abbreviate this by writing $\land$R even if we used the rule $\land$Rthree3 times and should have said $\land$R,$\land$R,$\land$R.

By sufficiently many uses of axiom $[]\wedge$, (7.18) and (7.19) are both equivalent to

$$A \to B \wedge [\alpha]B \wedge [\alpha][\alpha]B \wedge [\alpha][\alpha][\alpha]B \wedge [\alpha][\alpha][\alpha][\alpha][\alpha^*]B \tag{7.20}$$

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

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

If yes, we disproved (7.14), otherwise we use axiom $[^*]$ once more.

Note 39 (Bounded model checking) This process of iteratively unrolling a loop with the iteration axiom $[^*]$ and then checking the resulting (loop-free) conjuncts is called *Bounded Model Checking* and has been used with extraordinary success, e.g., in the context of finite-state systems [2]. The same principle can be useful to disprove properties of loops in differential dynamic logic by unwinding the loop, checking to see whether the resulting formulas have counterexamples and, if not, unrolling the loop once more. With certain computational refinements, this idea has found application in hybrid systems [1, 3, 5, 6] despite certain inevitable limits [9].

Suppose such a bounded model checking process has been followed to unroll the loop $N \in \mathbb{N}$ times. What can you conclude about the safety of the system?

If a counterexample is found or the formula can be disproved, then we are certain that the CPS is unsafe. If, instead, all but the last conjunct in the Nth unrolling of the loop are provable then the system will be safe for $N-1$ steps, but we cannot conclude anything about the safety of the system after more than $N-1$ steps. On the other hand, what we learn about the behavior of α from these iterations can still inform us about possible invariants.

7.7.2 Breaking Loops of Proofs

Proving properties of loops by unwinding them forever with axiom $[*]$ is not a promising strategy, unless we find that the conjecture is not valid after a number of unwindings. Or unless we do not mind being busy with the proof forever for infinitely many proof steps (which would never get the acrophobic bouncing ball off the ground either with the confidence that a safety argument provides). One way or another, we will have to find a way to break the loop apart to complete our reasoning.

How can we prove the premises of Fig. 7.6? Sect. 7.7.1 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 and allows us to get the proof over with after finitely many steps?

There is not all that much we can do to improve the way we prove the first premise ($A \vdash B$). We simply have to bite the bullet and do it, armed with all our knowledge of arithmetic from Chap. 6. But it's actually very easy at least for the

$$[*]\dfrac{[*]\dfrac{[*]\dfrac{\land\mathrm{R}\dfrac{A\vdash B \quad \mathrm{MR}\dfrac{A\vdash[\alpha]J_1 \quad \land\mathrm{R}\dfrac{J_1\vdash B \quad \mathrm{MR}\dfrac{J_1\vdash[\alpha]J_2 \quad \land\mathrm{R}\dfrac{J_2\vdash B \quad \dfrac{J_2\vdash[\alpha]J_3 \quad \dots}{J_2\vdash[\alpha][\alpha^*]B}}{J_2\vdash B\land[\alpha][\alpha^*]B}}{J_1\vdash[\alpha](B\land[\alpha][\alpha^*]B)}}{J_1\vdash B\land[\alpha](B\land[\alpha][\alpha^*]B)}}{A\vdash[\alpha]\big(B\land[\alpha](B\land[\alpha][\alpha^*]B)\big)}}{A\vdash B\land[\alpha]\big(B\land[\alpha](B\land[\alpha][\alpha^*]B)\big)}}{A\vdash B\land[\alpha](B\land[\alpha][\alpha^*]B)}}{A\vdash B\land[\alpha][\alpha^*]B}}{A\vdash[\alpha^*]B}$$

Fig. 7.6 Loops of proofs: iterating and generalizing the box

bouncing ball. Besides, no dynamics have 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 either, 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 dL 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 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 this intermediate condition E to simplify our proof?

Recall the intermediate condition contract version of the sequential composition proof rule from Chap. 4 that we briefly revisited in Chap. 5:

$$\mathrm{H;}\ \dfrac{A\rightarrow[\alpha]E \quad E\rightarrow[\beta]B}{A\rightarrow[\alpha;\beta]B}$$

Chap. 5 ended up dismissing the intermediate contract rule H; in favor of the more general axiom

$$[;]\quad [\alpha;\beta]P\leftrightarrow[\alpha][\beta]P$$

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

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

and then use rule H; to generalize with an intermediate condition E in the middle. However, this is what we generally want 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 axiom [;] forwards and then backwards and then forwards again and so on until the end of time. Such a looping proof does not strike us as useful. Instead, we'll adopt a proof rule that has some of the properties of H; but is more general. It is called *generalization* and allows us to prove any stronger postcondition Q for a modality, i.e., a postcondition that implies the original postcondition P.

Lemma 7.4 (MR monotonicity right rule). *This is a derived proof rule:*

$$\text{MR}\ \frac{\Gamma \vdash [\alpha]Q, \Delta \quad Q \vdash P}{\Gamma \vdash [\alpha]P, \Delta}$$

Proof. Rule MR can be derived from the monotonicity rule M[·] from Lemma 5.13:

$$\text{cut}\ \frac{\Gamma \vdash [\alpha]Q, \Delta \qquad \text{M}[\cdot]\ \dfrac{Q \vdash P}{\Gamma, [\alpha]Q \vdash [\alpha]P, \Delta}}{\Gamma \vdash [\alpha]P, \Delta}$$

□

Because the proof rule MR is just a cut away from monotonicity rule M[·], we will also just say that we prove by M[·] even if we really also used it together with a cut as in rule MR.

If we apply rule MR on the third premise $A \vdash [\alpha][\alpha]B$ of our bounded-model-checking-style proof attempt with the intermediate condition E for Q that we assume we have identified, then we end up with

$$\text{MR}\ \frac{A \vdash [\alpha]E \qquad E \vdash [\alpha]B}{A \vdash [\alpha][\alpha]B}$$

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

$$A \rightarrow B \wedge [\alpha](B \wedge [\alpha](B \wedge [\alpha](B \wedge [\alpha][\alpha^*]B))) \tag{7.18*}$$

Using rules ∧R and MR a number of times for a sequence of intermediate conditions E_1, E_2, E_3 derives the proof in Fig. 7.7.

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 (7.14) in the first place. But the derivation hints at a useful way we could possibly shortcut proofs. To lead to a proof of the conclusion, the above derivation requires us to prove the premises

$$\scriptstyle{\rightarrow\mathrm{R}}\dfrac{\scriptstyle{\wedge\mathrm{R}}\dfrac{A \vdash B \quad \scriptstyle{\mathrm{MR}}\dfrac{A \vdash [\alpha]E_1 \quad \scriptstyle{\wedge\mathrm{R}}\dfrac{E_1 \vdash B \quad \scriptstyle{\mathrm{MR}}\dfrac{E_1 \vdash [\alpha]E_2 \quad \scriptstyle{\wedge\mathrm{R}}\dfrac{E_2 \vdash B \quad \scriptstyle{\mathrm{MR}}\dfrac{E_2 \vdash [\alpha]E_3 \quad \scriptstyle{\wedge\mathrm{R}}\dfrac{E_3 \vdash B \quad E_3 \vdash [\alpha][\alpha^*]B}{E_3 \vdash B \wedge [\alpha][\alpha^*]B}}{E_2 \vdash [\alpha](B \wedge [\alpha][\alpha^*]B)}}{E_2 \vdash B \wedge [\alpha](B \wedge [\alpha][\alpha^*]B)}}{E_1 \vdash [\alpha](B \wedge [\alpha](B \wedge [\alpha][\alpha^*]B))}}{E_1 \vdash B \wedge [\alpha](B \wedge [\alpha](B \wedge [\alpha][\alpha^*]B))}}{A \vdash [\alpha](B \wedge [\alpha](B \wedge [\alpha](B \wedge [\alpha][\alpha^*]B)))}}{A \vdash B \wedge [\alpha](B \wedge [\alpha](B \wedge [\alpha](B \wedge [\alpha][\alpha^*]B)))}}{\vdash A \rightarrow B \wedge [\alpha](B \wedge [\alpha](B \wedge [\alpha](B \wedge [\alpha][\alpha^*]B)))}$$

Fig. 7.7 Loops of proofs: intermediate generalizations

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

as well as some other premises. What is an easy way to make that happen? What if all the intermediate conditions E_i were the same? Let's assume they are all the same condition E, that is, $E_1 \equiv E_2 \equiv E_3 \equiv E$. In that case, most of the resulting premises actually turn out to be one and the same premise:

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

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 iterations 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 $E \vdash [\alpha]E$ as one of the premises.

7.7.3 *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, no matter which of the states in E it was when the system started in the first place. It sounds like the system α^* cannot get out of E 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. So no matter how often our CPS repeats α^*, it will still reside in E.

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 a perfectly true invariant of the system, but not really a very useful one as far as proving B goes.

What else could go wrong in a system that obeys $E \vdash [\alpha]E$, i.e., where this sequent is valid, because we found a proof for it? Indeed, the other thing that could happen is that E is an invariant of the system that implies safety, but our system just does not initially start in E; then we still don't know whether it's safe. Taking all three conditions together, we arrive exactly at the loop induction rule from Lemma 7.3.

7.7.4 Alternative Forms of the Induction Axiom

In the literature [4, 7, 8, 10], the induction axiom is classically presented as

$$\text{II} \quad [\alpha^*](P \rightarrow [\alpha]P) \rightarrow (P \rightarrow [\alpha^*]P)$$

instead of the slightly stronger and more intuitive form developed in Sect. 7.3.1:

$$\text{I} \quad [\alpha^*]P \leftrightarrow P \wedge [\alpha^*](P \rightarrow [\alpha]P)$$

The classical axiom II is equivalent to the sufficiency direction "←" of the equivalence axiom I just by propositional rephrasing, because both axioms need both P and $[\alpha^*](P \rightarrow [\alpha]P)$ to imply $[\alpha^*]P$. The derivation of the necessity direction "→" of the equivalence axiom I from II needs a more elaborate argument.

The proof first derives the backwards iteration axiom from either the "←" necessity direction of induction axiom I (or axiom II) with the help of others.

Lemma 7.5 ($\overleftarrow{[*]}$ backwards iteration axiom). *This axiom is derived:*

$$\overleftarrow{[*]} \quad [\alpha^*]P \leftrightarrow P \wedge [\alpha^*][\alpha]P$$

Proof. The sufficiency direction "←" of axiom $\overleftarrow{[*]}$ directly derives from the sufficiency direction "←" of the induction axiom I or its equivalent classical axiom II using monotonicity rule M[·], because postcondition $[\alpha]P$ is stronger than $P \rightarrow [\alpha]P$:

$$\begin{prooftree}
\AxiomC{$*$}
\LeftLabel{id}
\UnaryInfC{$P \vdash P$}
\AxiomC{$*$}
\LeftLabel{id}
\UnaryInfC{$[\alpha]P, P \vdash [\alpha]P$}
\LeftLabel{$\rightarrow$R}
\UnaryInfC{$[\alpha]P \vdash P \rightarrow [\alpha]P$}
\LeftLabel{M$[\cdot]$}
\UnaryInfC{$[\alpha^*][\alpha]P \vdash [\alpha^*](P \rightarrow [\alpha]P)$}
\LeftLabel{$\wedge$R}
\BinaryInfC{$P, [\alpha^*][\alpha]P \vdash P \wedge [\alpha^*](P \rightarrow [\alpha]P)$}
\LeftLabel{$\wedge$L,I}
\UnaryInfC{$P \wedge [\alpha^*][\alpha]P \vdash [\alpha^*]P$}
\LeftLabel{$\rightarrow$R}
\UnaryInfC{$\vdash P \wedge [\alpha^*][\alpha]P \rightarrow [\alpha^*]P$}
\end{prooftree}$$

The necessity direction “→” of $\overleftarrow{[*]}$ is derived using [*],G,MR,[]∧ from axiom II or its equivalent sufficiency direction “←” of axiom I as shown in Fig. 7.8. □

$$\begin{prooftree}
\AxiomC{$*$}
\LeftLabel{id}
\UnaryInfC{$P, [\alpha][\alpha^*]P \vdash P$}
\LeftLabel{$\wedge$L}
\UnaryInfC{$P \wedge [\alpha][\alpha^*]P \vdash P$}
\LeftLabel{$[*]$}
\UnaryInfC{$[\alpha^*]P \vdash P$}
\AxiomC{$*$}
\UnaryInfC{$\vdash P \wedge [\alpha][\alpha^*]P \rightarrow [\alpha][\alpha^*]P$}
\LeftLabel{$[*]$}
\UnaryInfC{$\vdash [\alpha^*]P \rightarrow [\alpha][\alpha^*]P$}
\LeftLabel{G}
\UnaryInfC{$\vdash [\alpha^*]([\alpha^*]P \rightarrow [\alpha][\alpha^*]P)$}
\LeftLabel{II}
\UnaryInfC{$[\alpha^*]P \vdash [\alpha^*][\alpha^*]P$}
\AxiomC{$*$}
\LeftLabel{$\wedge$L,id}
\UnaryInfC{$[\alpha]P \wedge [\alpha][\alpha][\alpha^*]P \vdash [\alpha]P$}
\LeftLabel{$[]\wedge$}
\UnaryInfC{$[\alpha](P \wedge [\alpha][\alpha^*]P) \vdash [\alpha]P$}
\LeftLabel{$[*]$}
\UnaryInfC{$[\alpha][\alpha^*]P \vdash [\alpha]P$}
\LeftLabel{$\wedge$L,WL}
\UnaryInfC{$P \wedge [\alpha][\alpha^*]P \vdash [\alpha]P$}
\LeftLabel{$[*]$}
\UnaryInfC{$[\alpha^*]P \vdash [\alpha]P$}
\LeftLabel{MR}
\BinaryInfC{$[\alpha^*]P \vdash [\alpha^*][\alpha]P$}
\LeftLabel{$\wedge$R}
\BinaryInfC{$[\alpha^*]P \vdash P \wedge [\alpha^*][\alpha]P$}
\LeftLabel{$\rightarrow$R}
\UnaryInfC{$\vdash [\alpha^*]P \rightarrow P \wedge [\alpha^*][\alpha]P$}
\end{prooftree}$$

Fig. 7.8 Derivation of backwards unwinding axiom from alternative induction axiom

With the help of the backwards iteration axiom $\overleftarrow{[*]}$, the proof of the necessity direction “→” of axiom I is straightforward, because the only difference is the additional assumption P in the postcondition:

$$\begin{prooftree}
\AxiomC{$*$}
\LeftLabel{id}
\UnaryInfC{$P \vdash P$}
\AxiomC{$*$}
\LeftLabel{id}
\UnaryInfC{$[\alpha]P, P \vdash [\alpha]P$}
\LeftLabel{$\rightarrow$R}
\UnaryInfC{$[\alpha]P \vdash P \rightarrow [\alpha]P$}
\LeftLabel{M$[\cdot]$}
\UnaryInfC{$[\alpha^*][\alpha]P \vdash [\alpha^*](P \rightarrow [\alpha]P)$}
\LeftLabel{$\wedge$L,WL}
\UnaryInfC{$P \wedge [\alpha^*][\alpha]P \vdash [\alpha^*](P \rightarrow [\alpha]P)$}
\LeftLabel{$\overleftarrow{[*]}$}
\UnaryInfC{$[\alpha^*]P \vdash [\alpha^*](P \rightarrow [\alpha]P)$}
\LeftLabel{$\wedge$R}
\BinaryInfC{$[\alpha^*]P \vdash P \wedge [\alpha^*](P \rightarrow [\alpha]P)$}
\LeftLabel{$\rightarrow$R}
\UnaryInfC{$\vdash [\alpha^*]P \rightarrow P \wedge [\alpha^*](P \rightarrow [\alpha]P)$}
\end{prooftree}$$

This completes the derivation of the induction axiom I from its classical formulation II using the other axioms.

This proof proved $[\alpha^*]P \rightarrow [\alpha^*][\alpha^*]P$, which has a stronger formulation.

Lemma 7.6 ([**] double iteration axiom). *This axiom is derived:*

$$[**] \quad [\alpha^*;\alpha^*]P \leftrightarrow [\alpha^*]P$$

Proof. The composition axiom [;] reduces the proof to two directions of which the direction "←" was already proved in the middle branch of Fig. 7.8:

$$
{\scriptstyle [;]}\dfrac{{\scriptstyle \leftrightarrow\mathrm{R}}\dfrac{{\scriptstyle [*]}\dfrac{{\scriptstyle \wedge\mathrm{L,id}}\dfrac{*}{[\alpha^*]P\wedge[\alpha][\alpha^*][\alpha^*]P \vdash [\alpha^*]P}}{[\alpha^*][\alpha^*]P \vdash [\alpha^*]P} \qquad \dfrac{*\ (\text{Fig. } 7.8)}{[\alpha^*]P \vdash [\alpha^*][\alpha^*]P}}{\vdash [\alpha^*][\alpha^*]P \leftrightarrow [\alpha^*]P}}{\vdash [\alpha^*;\alpha^*]P \leftrightarrow [\alpha^*]P}
$$

□

Derived axiom [**] is mostly meant to be used from left to right in order to collapse two subsequent loops into a single loop. Its diamond modality counterpart can also be useful to split a loop $\langle\alpha^*\rangle P$ into two separate loops $\langle\alpha^*;\alpha^*\rangle P$:

$$\langle{**}\rangle \quad \langle\alpha^*;\alpha^*\rangle P \leftrightarrow \langle\alpha^*\rangle P$$

This splitting can be useful, e.g., to show that, on an empty soccer field, a robot can kick the ball into the goal in a control loop. After duplicating the control loop by axiom $\langle{**}\rangle$, it is sufficient to show that the first control loop can navigate the robot close enough to the goal to have a chance to score, and then the second control loop can repeat until a goal is scored.

Exercises

7.1. Give a sequent proof for:

$$2gx = 2gH - v^2 \wedge x \geq 0 \rightarrow [x' = v, v' = -g \,\&\, x \geq 0](2gx = 2gH - v^2 \wedge x \geq 0)$$

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

$$2gx = 2gH - v^2 \wedge x \geq 0 \rightarrow [x' = v, v' = -g](2gx = 2gH - v^2 \wedge x \geq 0)$$

7.2. Section 7.3.1 argued that both P and $[\alpha]P$ follow from (7.3). Show how $[\alpha]P$ is, indeed, implied by applying the iteration axiom [*] one more time to (7.3).

7.3 (Invariant candidates for bouncing balls). Could the bouncing ball use any of the following formulas as invariants to prove (7.1)? Explain why.

$$\mathrm{j}(x,v) \overset{\text{def}}{\equiv} (x = 0 \vee x = H) \wedge v = 0$$
$$\mathrm{j}(x,v) \overset{\text{def}}{\equiv} 0 \leq x \wedge x \leq H \wedge v^2 \leq 2gH$$
$$\mathrm{j}(x,v) \overset{\text{def}}{\equiv} 0 \leq x \wedge x \leq H \wedge v \leq 0$$

7.4. Conduct a sequent proof for (7.12) without the monotonicity rule MR.

7.5 (Damped bouncing balls). Section 7.4 proved the bouncing-ball formula (7.12) in the dL sequent calculus for the case $c = 1$ of no damping at the bounce. Putting aside Quantum, actual bouncing balls are less perfect and only achieve damping coefficients satisfying $0 \leq c \leq 1$. Identify a suitable invariant and conduct a sequent calculus proof for this generalization.

7.6. Identify loop invariants proving the following dL formulas:

$$x > 1 \rightarrow [(x:=x+1)^*]x \geq 0$$

$$x > 5 \rightarrow [(x:=2)^*]x > 1$$

$$x > 2 \land y \geq 1 \rightarrow [(x:=x+y;y:=y+2)^*]x > 1$$

$$x > 2 \land y \geq 1 \rightarrow [(x' = y)^*]x > 1$$

$$x > 2 \land y \geq 1 \rightarrow [(x:=y;x' = y)^*]x \geq 1$$

$$x = -1 \rightarrow [(x:=2x+1)^*]x \leq 0$$

$$x = -1 \rightarrow [(\{x' = 2\})^*]x \geq -5$$

$$x = 5 \land c > 1 \land d > -c \rightarrow [(\{x' = c+d\})^*]x \geq 0$$

$$x = 1 \land u > x \rightarrow [\big(x:=2;\{x' = x^2 + u\}\big)^*]x \geq 0$$

$$x = 1 \land y = 2 \rightarrow [(x:=x+1;\{x' = y, y' = 2\})^*]x \geq 0$$

$$x \geq 1 \land v \geq 0 \rightarrow [\big(\{x' = v, v' = 2\}\big)^*]x \geq 0$$

$$x \geq 1 \land v > 0 \land A > 0 \rightarrow [\big((a:=0 \cup a:=A);\{x' = v, v' = a\}\big)^*]x \geq 0$$

7.7. Give a direct semantic soundness proof for rule ind and contrast its soundness proof with the soundness proof of rule loop to observe similarities and differences.

7.8 (Constant parameter assumptions). As Example 7.3 showed, it would be unsound for either the loop invariant rule or its core counterpart rule ind to keep the context Γ, Δ in the induction step (or in the third premise of rule loop). With adequate care, *some* select formulas from Γ and Δ can still be kept without losing soundness. These are the formulas from Γ and Δ that only refer to constant parameters that do not change during the HP α^*. Give a semantic argument why such a constant formula q in Γ or Δ can be kept soundly. Then show how q can be retained in the induction step with the help of the vacuous axiom V from Sect. 5.6.2.

7.9 (Far induction). The far induction axiom is for quicker inductions since its induction step takes two steps at once. Is it sound? Prove that it is or show a counterexample.

$$[\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha][\alpha]P)$$

7.10 (Unwound). The appendix motivated the loop invariant proof rule via systematic unwinding considerations in Sect. 7.7.2. We unwound loops in two different ways either directly in Fig. 7.6 or with intermediate generalizations in Fig. 7.7. Both approaches ultimately took us to the same inductive principle. But if unwinding is all that we are interested in, then which of the two ways of unwinding is more efficient? Which one produces feweer premises that are distractions in the argument? Which one has fewer choices of different intermediate conditions E_i in the first place?

7.11 (First arrival). Show that the first arrival axiom is sound, which says that if P is reachable by repeating α, then P is either *true* right away or one can repeat P to a state where P is not *true* yet but will become *true* after the next iteration of α:

$$\text{FA} \quad \langle\alpha^*\rangle P \to P \lor \langle\alpha^*\rangle(\neg P \land \langle\alpha\rangle P)$$

7.12 (*Dribbling basket balls). Identify a requirement on the initial state of the bouncing ball that allows it to move initially, so is more general than $v = 0$. Prove that this variation of the bouncing ball is safe.

References

[1] Alessandro Cimatti, Sergio Mover, and Stefano Tonetta. SMT-based scenario verification for hybrid systems. *Formal Methods in System Design* **42**(1) (2013), 46–66. DOI: 10.1007/s10703-012-0158-0.

[2] Edmund M. Clarke, Armin Biere, Richard Raimi, and Yunshan Zhu. Bounded model checking using satisfiability solving. *Form. Methods Syst. Des.* **19**(1) (2001), 7–34. DOI: 10.1023/A:1011276507260.

[3] Andreas Eggers, Martin Fränzle, and Christian Herde. SAT modulo ODE: a direct SAT approach to hybrid systems. In: *Automated Technology for Verification and Analysis, 6th International Symposium, ATVA 2008, Seoul, Korea, October 20-23, 2008. Proceedings*. Ed. by Sung Deok Cha, Jin-Young Choi, Moonzoo Kim, Insup Lee, and Mahesh Viswanathan. Vol. 5311. LNCS. Berlin: Springer, 2008, 171–185. DOI: 10.1007/978-3-540-88387-6_14.

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

[5] Soonho Kong, Sicun Gao, Wei Chen, and Edmund M. Clarke. dReach: δ-reachability analysis for hybrid systems. In: *Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*. Ed. by Christel Baier and Cesare Tinelli. Vol. 9035. LNCS. Berlin: Springer, 2015, 200–205.

[6] Carla Piazza, Marco Antoniotti, Venkatesh Mysore, Alberto Policriti, Franz Winkler, and Bud Mishra. Algorithmic algebraic model checking I: challenges from systems biology. In: *CAV*. Ed. by Kousha Etessami and Sriram K. Rajamani. Vol. 3576. LNCS. Berlin: Springer, 2005, 5–19. DOI: 10.1007/11513988_3.

[7] André Platzer. The complete proof theory of hybrid systems. In: *LICS*. Los Alamitos: IEEE, 2012, 541–550. DOI: 10.1109/LICS.2012.64.

[8] André Platzer. A complete uniform substitution calculus for differential dynamic logic. *J. Autom. Reas.* **59**(2) (2017), 219–265. DOI: 10.1007/s10817-016-9385-1.

[9] André Platzer and Edmund M. Clarke. The image computation problem in hybrid systems model checking. In: *HSCC*. Ed. by Alberto Bemporad, Antonio Bicchi, and Giorgio C. Buttazzo. Vol. 4416. LNCS. Springer, 2007, 473–486. DOI: 10.1007/978-3-540-71493-4_37.

[10] Vaughan R. Pratt. Semantical considerations on Floyd-Hoare logic. In: *17th Annual Symposium on Foundations of Computer Science, 25-27 October 1976, Houston, Texas, USA*. Los Alamitos: IEEE, 1976, 109–121. DOI: 10.1109/SFCS.1976.27.