André Platzer

Foundations of Cyber-Physical Systems

– Textbook –

Springer

Chapter 12
Ghosts & Differential Ghosts

Synposis While working toward yet another fundamental reasoning technique for differential equations, this chapter describes somewhat surprising uses of additional auxiliary variables, called ghosts, in the modeling of and reasoning about cyber-physical systems. Upon closer inspection, there are a number of reasons to include minor ghost variables for simpler purposes that have so far gone unnoticed. But extra ghost variables can be leveraged in surprising ways, even to explain how differential equations can be solved as part of an ordinary differential induction proof. The chapter culminates in the insight that even differential equations themselves need ghosts, which are called differential ghosts and can make a surprising difference in our understanding.

12.1 Introduction

Chapters 10 and 11 equipped us with powerful tools for proving properties of differential equations without having to solve them. *Differential invariants* (dI) [4, 6] prove properties of differential equations by induction based on the right-hand side of the differential equation, rather than its much more complicated global solution. *Differential cuts* (dC) [4, 6] made it possible to simply prove another property C of a differential equation and then change the dynamics of the system around so that it is restricted to never leave region C. Differential cuts are a fundamental proof principle that can make properties inductive that are not otherwise invariants [6]. They do so by soundly changing the evolution domain constraints.

Yet, not every true property of a differential equation can be proved even with the help of differential cuts [6]. Indeed, there is another way of transforming the dynamics of the system in ways that can enable new proofs that were not possible before [6]. This transformation uses *differential ghosts* [6, 8] to soundly change the differential equations themselves instead of just changing its evolution domain constraints like differential cuts do. Of course, editing the differential equations should make us

even more nervous about soundness than editing the evolution domain already did in Chap. 11.

Differential ghosts are extra variables that are introduced into the differential equation system solely for the purpose of the proof. This existence just for analytic purposes is where the spooky name "ghost" comes from. Ghosts (or auxiliaries) refer to aspects of a model that do not exist in reality but are merely introduced for the sake of its analysis. Ghosts are not really present in the actual system, but just invented to make the story more interesting or, rather, the proof more conclusive.

In fact, ghost variables can also be useful for a proof when they remain entirely discrete variables that only change by discrete assignments, in which case they are called *discrete ghosts*. Such discrete ghosts are used to remember an intermediate state during the execution, which makes it possible to conduct a proof that relates the new value to the old value stored in the discrete ghost. Why would that be useful? Well, because it is sometimes easier to analyze the change of a variable than the value of the variable itself.

Both discrete ghosts and differential ghosts serve a similar intuitive purpose: remember intermediate state values so that the relation of the values at intermediate states to values at final states can be analyzed. The difference is that differential ghosts also update their value continuously at will along their very own special differential equation, which, if cleverly chosen, makes it particularly easy to conduct a proof. Ghosts give the proof a way of referring to how the state used to be that is no more. There are many reasons for introducing ghost state into a system, which will be investigated in this chapter. One particularly intuitive motivation for introducing a differential ghost is for proofs of properties that get less true over time, such that invariance techniques alone cannot prove them. A particularly cleverly chosen differential ghost can serve as a counterweight to the change of truth of the original postcondition, which will serve as an (evolving) point of reference, e.g., when the rate of change of the truth value is changing over time.

The results in this chapter are loosely based on [6, 8].

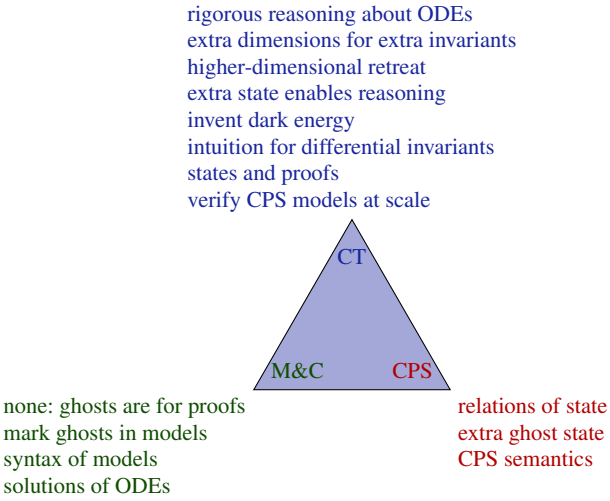
The most important learning goals of this chapter are:

Modeling and Control: This chapter does not have much impact on modeling and control of CPS, because, after all, the whole point of ghosts and differential ghosts is that they are only added for the purposes of the proof. However, it can still sometimes be helpful to add such ghost and differential ghost variables into the original model right away. It is good style to mark such additional variables in the model and controller as ghost variables in order to retain the fact that they do not need to be included in the final system executable except for monitoring.

Computational Thinking: This chapter leverages computational thinking principles for the purpose of rigorous reasoning about CPS models by analyzing how extra dimensions can simplify or enable reasoning about lower-dimensional systems. From a state space perspective, extra dimensions are a horrible idea, because, e.g., the number of points on a gridded space grows exponentially in the number of dimensions (curse of dimensionality). From a reasoning perspective, however, the important insight of this chapter is that extra state variables sometimes help and may even make reasoning possible that is otherwise impossible.

One intuition why extra ghost state may help reasoning is that it can be used to consume the energy that a given dissipative system is leaking (similar to the purpose why dark matter had been speculated to exist) or produce the energy that a given system consumes. The addition of such extra ghost state then enables invariants of generalized energy constants involving both original and ghost state that was not possible using only the original state. That is, ghost state may cause new energy invariants. This chapter continues the trend of generalizing important logical phenomena from discrete to continuous systems. The verification techniques developed in this chapter are critical for verifying some CPS models of appropriate scale and technical complexity but are not necessary for all CPS models. A secondary goal of this chapter is to develop more intuition and deeper understandings of differential invariants and differential cuts.

CPS Skills: The focus in this chapter is on reasoning about CPS models, but there is an indirect impact on developing better intuitions for operational effects in CPS by introducing the concept of relations of state to extra ghost state. A good grasp on such relations can substantially help with the intuitive understanding of CPS dynamics. The reason is that ghosts and differential ghosts enable extra invariants, which enable stronger statements about what we can rely on as a CPS evolves. The also enable relational arguments of how the change of some quantity over time relates to the change of another auxiliary quantity.



12.2 Recap

Recall the following proof rules for differential invariants (dI), differential weakening (dW) and differential cuts (dC) for differential equations from Chap. 11:

Note 62 (Proof rules for differential equations)

$$\text{dI}\ \frac{Q \vdash [x':=f(x)](F)'}{F \vdash [x'=f(x)\,\&\,Q]F} \qquad \text{dW}\ \frac{Q \vdash P}{\Gamma \vdash [x'=f(x)\,\&\,Q]P,\Delta}$$

$$\text{dC}\ \frac{\Gamma \vdash [x'=f(x)\,\&\,Q]C,\Delta \quad \Gamma \vdash [x'=f(x)\,\&\,(Q\wedge C)]P,\Delta}{\Gamma \vdash [x'=f(x)\,\&\,Q]P,\Delta}$$

12.3 A Gradual Introduction to Ghost Variables

This section provides a gradual introduction into various forms of ghost variables. It focuses on the motivation and intuition for what ghost variables are for and how they can help conducting a proof.

12.3.1 Discrete Ghosts

The discrete way of adding ghost variables is to introduce a new ghost variable y into a proof that remembers the value of any arbitrary term e for later usage. This can be useful in a proof in order to have a name, y, that recalls the value of e later on in the proof, especially when the value of e changes subsequently during the execution of hybrid programs α in the remaining modalities. Such a discrete ghost y makes it possible to relate the value of e before and after the run of that hybrid program α.

Lemma 12.1 (Discrete ghosts). *The following is a sound proof rule for introducing an auxiliary variable or (discrete)* ghost *y:*

$$\text{IA}\ \frac{\Gamma \vdash [y:=e]p,\Delta}{\Gamma \vdash p,\Delta}\quad (y \text{ new})$$

Proof. Rule IA derives from assignment axiom $[:=]$ from Chap. 5, which proves

$$p \leftrightarrow [y:=e]p$$

because the new variable y does not occur in p (which can be thought of as a nullary predicate symbol here). □

The discrete ghost rule IA directly derives from the assignment axiom [:=] since it merely applies the assignment axiom backwards to introduce a ghost variable y that was not there before. This is an example exploiting the flexibility of equivalence axioms to be used forward as well as backwards. Of course, it is important to retain a forward momentum in the proof and not apply assignment axiom [:=] to the premise of IA, which would make the carefully dreamt up discrete ghost y go away again:

$$\text{IA}\,\dfrac{\dfrac{\Gamma \vdash p, \Delta}{[:=]\;\dfrac{\Gamma \vdash [y := e]p, \Delta}{}}}{\Gamma \vdash p, \Delta}$$

Making ghosts go away would be a great goal for the Ghostbusters, but would not exactly make proof attempts productive. If we were really planning on eliminating the discrete ghost again, then we should never have introduced it with rule IA in the first place. After observing that discrete ghosts are fancy names for backwards assignments and, thus, sound, the next question is what they could possibly be good for.

Discrete ghosts can be interesting when p in rule IA contains modalities that change variables in term e so that y can remember the value that e had before that change. For example:

$$\text{IA}\,\dfrac{xy \geq 2 \vdash [c := xy][x' = x, y' = -y]\, xy \geq}{xy \geq 2 \vdash [x' = x, y' = -y]\, xy \geq 2}$$

This proof memorizes in the discrete ghost variable c the value that the interesting term xy had before the differential equation started. It is not obvious how to complete the proof, because substituting c away using the assignment axiom [:=] would undo the pleasant effect that rule IA had, because the whole point of the new variable c is that it does not occur elsewhere.[1] So the only way the proof can make progress is by applying a proof rule to the differential equation, which is not top-level. We can either just keep the assignment around and directly use axioms at the postcondition. Or we first turn the assignment into an equation with this derived proof rule.

Lemma 12.2 ($[:=]_=$R Equational assignment). *This is a derived rule:*

$$[:=]_=\text{R}\;\dfrac{\Gamma, y = e \vdash p(y), \Delta}{\Gamma \vdash [x := e]p(x), \Delta} \quad (y \text{ new})$$

Proof. The derivation is shown in prior work [8, Theorem 40]. □

[1] This potentially surprising phenomenon happens in some form or other for other ghosts as well, because, the whole point of ghosts is to compute something that the original model and property do not depend on. So, sufficiently sophisticated forms of dead-code elimination would get rid of ghosts, which would be counterproductive for the proof. In fact, dead-code elimination for compilers and ghosts for proofs are the same phenomenon. Only backwards, because, applied from bottom to top, the discrete ghost rule IA introduces a variable that is dead code as opposed to eliminating it.

With that rule we can proceed as if nothing had happened:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{\vdash 0=xy+x(-y)}\,\mathbb{R}}{\vdash [x':=x][y':=-y]0=x'y+xy'}\,[:=]}{xy\geq 2, c=xy \vdash [x'=x,y'=-y]c=xy}\,dI \quad \triangleright}{xy\geq 2, c=xy \vdash [x'=x,y'=-y]xy\geq 2}\,MR}{xy\geq 2 \vdash [c:=xy][x'=x,y'=-y]xy\geq 2}\,[:=]_=R}{xy\geq 2 \vdash [x'=x,y'=-y]xy\geq 2}\,IA$$

The generalization step MR leads to a second premise that has been elided (marked by $\triangleright$) and proves, e.g., by vacuous axiom V, because the discrete ghost c starts out above 2 by antecedent and never changes its value during the differential equation. This particular property also proves directly quite easily, but the proof technique of discrete ghosts is of more general interest beyond this demonstration. The next section provides one common source of such examples.

Notice that even the initial height H of the bouncing ball model in Sect. 4.5 could have been considered a discrete ghost for the purpose of remembering the initial height via $H:=x$ initially. The only reason why it is not a pure discrete ghost is because it is also used in the postcondition, so the safety conjecture cannot be stated without H. The variable H is part of the property, not just part of the proof.

12.3.2 Proving Bouncing Balls with Sneaky Solutions

Recall the dL formula for a falling ball in the bouncing ball proof from Chap. 7:

$$2gx=2gH-v^2 \wedge x\geq 0 \rightarrow [\{x'=v,v'=-g \,\&\, x\geq 0\}](2gx=2gH-v^2 \wedge x\geq 0) \quad (11.7^*)$$

This formula was already proved twice. Once in Chap. 7 using the solution axiom schema $[']$ and once in Sect. 11.10 on p. 335 using a mix of differential invariants with differential weakening, because the postcondition has been cleverly constructed as an invariant of the bouncing ball in Chap. 4 already.

It's a good idea to be clever! But it also pays off to be systematic and develop a rich toolbox of techniques for proving properties of differential equations. Is there a way to prove (11.7) without such a distinctively clever invariant that works as a differential invariant right away? Yes, of course, there is one, because (11.7) can even be proved using solution axiom $[']$. How many proofs does a formula need these days until we stop proving it?

Well, of course, every formula only needs one proof and then lives happily valid ever after. But it turns out that interesting things happen when we systematically try to understand how to make a proof happen that does not use the solution axiom $[']$ and, yet, still uses solution-based arguments. Can you conceive a way to use solutions for differential equations without invoking the actual solution axiom $[']$?

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

The solution of a differential equation should be invariant along the differential equation, because it describes an identity that always holds when following the differential equation. The solution for balls falling in gravity according to (11.7) is:

$$x(t) = x + vt - \frac{g}{2}t^2$$
$$v(t) = v - gt$$

where x denotes the initial position and v_0 the initial velocity while $x(t)$ and $v(t)$ denote the position and velocity, respectively, after duration t. Now, the only trouble is that these equations cannot possibly be used as straight-out differential invariants, because $x(t)$ is not even allowed in the language we considered so far.[2] After the differential equation, the name for the position at that time is simply x and the name for velocity is just v. Obviously, $v = v - gt$ would not be a very meaningful equation as time moves on, so we somehow need to identify a new name for the old value that the position initially had before the differential equation, and likewise for velocity. This leads to the following rephrasing of the solution where x and v denote the variables after the differential equation at time t and x_0 and v_0 those before:

$$\begin{aligned} x &= x_0 + v_0 t - \frac{g}{2}t^2 \\ v &= v_0 - gt \end{aligned} \tag{12.1}$$

These equations are legitimate formulas and could possibly be differentially cut into (11.7) by dC, but there are still some nuanced subtleties with that approach.

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

Even if we mentioned that we intend v_0 to mean the initial value of velocity before the differential equation, there is no way that the proof will know this unless we do something about it. In particular, the proof will fail, because the resulting arithmetic is not true for all values of x_0 and v_0. Fortunately, there is a perfect proof rule that fits to the task as if it had been made for this job. Before handling the differential equation, the proof rule IA can introduce a discrete ghost x_0 that remembers the initial value of x in the new discrete ghost variable x_0. And the rule IA can be used again to remember the initial value of v in the discrete ghost v_0. From then on, the variables x_0 and v_0 really are the initial values of x and v before the ODE.

Now that the proof is equipped with a way of referring to the initial values, the next question is how to exactly go about differentially cutting the solutions (12.1) into the differential equations by dC. Maybe the most immediate suggestion would be to use rule dC with a conjunction of the two equations in (12.1) to get the solution into the system as quickly as possible. That will not work, however, because $x =$

[2] Function symbol applications like $x(t)$ will enter dL officially in Chap. 18, but that does not alter the considerations we are about to make.

$x_0 + v_0 t - \frac{g}{2}t^2$ only is the correct solution for $x' = v$ after we have established that $v = v_0 - gt$ also is the correct solution of the differential equation $v' = -g$ that x depends on.

In retrospect, this ordering of differential cuts makes sense, because the differential equation $x' = v, v' = -g$ explicitly indicates that the change of x depends on v whose change, in turn, depends on g, which remains constant. Consequently, we first need to convey with a differential cut what the behavior of v is before we proceed with investigating the behavior of x that, after all, depends on v.

Now, we are ready for a proof with solutions that does not use the solution axiom schema $[']$. In order to prevent us from accidentally being too clever and exploiting pure differential invariance principles again as in Sect. 11.10, we will pretend not to know anything about the specific precondition and postcondition and just call then A and $B_{(x,v)}$. Consider the following formulation of the dL formula (11.7):

$$A \vdash [\{x' = v, v' = -g, t' = 1 \,\&\, x \geq 0\}]B_{(x,v)} \tag{12.2}$$

$$\text{where} \quad A \overset{\text{def}}{\equiv} 2gx = 2gH - v^2 \wedge x \geq 0$$

$$B_{(x,v)} \overset{\text{def}}{\equiv} 2gx = 2gH - v^2 \wedge x \geq 0$$

$$\{x'' = -g, t' = 1\} \overset{\text{def}}{\equiv} \{x' = v, v' = -g, t' = 1\}$$

The proof begins by introducing a discrete ghost v_0 remembering the initial velocity of the bouncing ball and then differentially cuts the solution $v = v_0 - tg$ into the system which is proved to be differentially invariant:

$$\text{IA}\,\dfrac{\text{dC}\,\dfrac{\text{dI}\,\dfrac{[:=]\,\dfrac{\mathbb{R}\,\dfrac{*}{x \geq 0 \vdash -g = -1g}}{x \geq 0 \vdash [v' := -g][t' := 1]v' = -t'g}}{A \vdash [v_0 := v][x'' = -g, t' = 1 \,\&\, x \geq 0]v = v_0 - tg} \quad A \vdash [v_0 := v][x'' = -g, t' = 1 \,\&\, x \geq 0 \wedge v = v_0 - tg]B_{(x,v)}}{A \vdash [v_0 := v][x'' = -g, t' = 1 \,\&\, x \geq 0]B_{(x,v)}}}{A \vdash [x'' = -g, t' = 1 \,\&\, x \geq 0]B_{(x,v)}}$$

Observe how the differential invariant rule dI made the sequent context A as well as the assignment $[v_0 := v]$ disappear, which is important for soundness, because both only hold in the initial state. That both are affected is easy to see if we had turned the assignment $v_0 := v$ into an equation $v_0 = v$ with rule $[:=]_=$R. Besides, if v_0 is the initial value of v then v_0 does not remain equal to v as the ball falls along $v' = -g$.

The left premise in the above proof proved by trivial arithmetic (rule $\mathbb{R}$). The right premise in the above proof proves as follows by first introducing yet another discrete ghost x_0 with IA that remembers the initial position so that it can be referred to in the solution. The solution $x = x_0 + v_0 t - \frac{g}{2}t^2$ can then be differentially cut into the system by dC and proved to be differentially invariant by dI using the evolution domain $v = v_0 - tg$:

$$\text{id}\,\frac{*}{x\geq 0\wedge v=v_0-tg\vdash v=v_0-2\frac{g}{2}t}$$
$$[:=]\,\frac{x\geq 0\wedge v=v_0-tg\vdash [x':=v][t':=1]x'=v_0t'-2\frac{g}{2}tt'}{x\geq 0\wedge v=v_0-tg\vdash [x':=v][t':=1]x'=v_0t'-2\frac{g}{2}tt'}$$
$$\text{dI}\,\frac{}{A\vdash [x_0:=x][v_0:=v][x''=-g,t'=1\,\&\,x\geq 0\wedge v=v_0-tg]x=x_0+v_0t-\frac{g}{2}t^2\rhd}$$
$$\text{dC}\,\frac{}{A\vdash [x_0:=x][v_0:=v][x''=-g,t'=1\,\&\,x\geq 0\wedge v=v_0-tg]B_{(x,v)}}$$
$$\text{IA}\,\frac{}{A\vdash [v_0:=v][x''=-g,t'=1\,\&\,x\geq 0\wedge v=v_0-tg]B_{(x,v)}}$$

The differential cut proof step (dC) has a second premise using the cut which is elided above (marked by $\rhd$) and proves by differential weakening (dW):

$$\text{dW}\,\frac{x\geq 0\wedge v=v_0-tg\wedge x=x_0+v_0t-\frac{g}{2}t^2\vdash B_{(x,v)}}{A\vdash [x_0:=x][v_0:=v][x''=-g,t'=1\,\&\,x\geq 0\wedge v=v_0-tg\wedge x=x_0+v_0t-\frac{g}{2}t^2]B_{(x,v)}}$$

After expanding $B_{(x,v)}$, the resulting arithmetic can be proved by real arithmetic, but it has a twist! First of all, the arithmetic can be simplified substantially using the equality substitution rule =R from Chap. 6 to replace v by v_0-tg and replace x by $x_0+v_0t-\frac{g}{2}t^2$ and use subsequent weakening (WL) to get rid of both equations after use. This simplification reduces the computational complexity of real arithmetic:

$$\wedge\text{L}\,\frac{\text{WL}\,\frac{\vdash 2g(x_0+v_0t-\frac{g}{2}t^2)=2gH-(v_0-tg)^2}{x\geq 0\vdash 2g(x_0+v_0t-\frac{g}{2}t^2)=2gH-(v_0-tg)^2}\quad \wedge\text{R}\,\frac{\text{id}\,\frac{*}{x\geq 0\vdash x\geq 0}}{x\geq 0\vdash 2g(x_0+v_0t-\frac{g}{2}t^2)=2gH-(v_0-tg)^2\wedge x\geq 0}}{x\geq 0\wedge v=v_0-tg\wedge x=x_0+v_0t-\frac{g}{2}t^2\vdash 2gx=2gH-v^2\wedge x\geq 0}$$
$$\text{WL}\,\frac{x\geq 0\vdash 2g(x_0+v_0t-\frac{g}{2}t^2)=2gH-(v_0-tg)^2\wedge x\geq 0}{x\geq 0,v=v_0-tg,x=x_0+v_0t-\frac{g}{2}t^2\vdash 2g(x_0+v_0t-\frac{g}{2}t^2)=2gH-(v_0-tg)^2\wedge x\geq 0}$$
$$=\text{R}\,\frac{x\geq 0,v=v_0-tg,x=x_0+v_0t-\frac{g}{2}t^2\vdash 2gx=2gH-(v_0-tg)^2\wedge x\geq 0}{x\geq 0,v=v_0-tg,x=x_0+v_0t-\frac{g}{2}t^2\vdash 2gx=2gH-(v_0-tg)^2\wedge x\geq 0}$$
$$=\text{R}\,\frac{x\geq 0,v=v_0-tg,x=x_0+v_0t-\frac{g}{2}t^2\vdash 2gx=2gH-v^2\wedge x\geq 0}{x\geq 0,v=v_0-tg,x=x_0+v_0t-\frac{g}{2}t^2\vdash 2gx=2gH-v^2\wedge x\geq 0}$$

Observe how this use of equality substitution and weakening helped simplify the arithmetic complexity of the formula substantially and even helped to eliminate a variable (v) right away. This can be useful to simplify arithmetic in many other cases as well. Both eliminating variables as well as applying and hiding equations right away can often simplify the complexity of handling real arithmetic. The arithmetic in the remaining left branch

$$2g\left(x_0+v_0t-\frac{g}{2}t^2\right)=2gH-(v_0-tg)^2$$

expands by polynomial arithmetic and cancels as indicated:

$$2g\left(x_0+\cancel{v_0t}-\cancel{\frac{g}{2}t^2}\right)=2gH-v_0^2+\cancel{2v_0tg}+\cancel{t^2g^2}$$

Those cancellations simplify the arithmetic, leaving the remaining condition:

$$2gx_0=2gH-v_0^2 \tag{12.3}$$

Indeed, this relation characterizes exactly how H, which turns out to have been the maximal height, relates to the initial height x_0 and initial velocity v_0. In the case of initial velocity $v_0=0$, for example, the equation (12.3) collapses to $x_0=H$, i.e. that H is the initial height in that case. Consequently, the computationally fastest way of

proving the resulting arithmetic is to first prove by a differential cut dC that (12.3) is a trivial differential invariant (even by the vacuous axiom V), resulting in a proof of (11.7); see Exercise 12.3.

Yet, as we go through all proof branches again to check that we really have a proof, however, we notice a subtle but blatant oversight. Can you spot it, too?

The very first left-most branch with the initial condition for the differential invariant $v = v_0 = tg$ does not, actually, prove. The catch is that we silently assumed $t = 0$ to be the initial value for the new clock t, but our proof did not actually say so. Oh my, what could possibly be done about that glitch?

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

There are multiple approaches and, in fact, the most elegant approach will have to wait till the next section. But one feature that we have just learned about can be exploited again to talk about the change of time without having to assume that time t starts at 0. Discrete ghosts to the rescue! Even though we do not know the initial value of the differential ghost t, we can simply use a discrete ghost again to call it t_0 and get on with it. Will that work? Can you find it out? Or should we start a revision of the proof to find out?

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{x\geq 0 \vdash -g=-1g}\,\mathbb{R}}{x\geq 0 \vdash [v':=-g][t':=1]v'=0-(t'-0)g}\,[:=]}{A \vdash [t_0:=t][v_0:=v][x''=-g,t'=1\,\&\,x\geq 0]v=v_0-(t-t_0)g\ \triangleright}\,\text{dI}}{A \vdash [t_0:=t][v_0:=v][x''=-g,t'=1\,\&\,x\geq 0]B_{(x,v)}}\,\text{dC}}{A \vdash [v_0:=v][x''=-g,t'=1\,\&\,x\geq 0]B_{(x,v)}}\,\text{IA}$$

The proof continues similarly with this elided premise (marked $\triangleright$ above):

$$A \vdash [t_0:=t][v_0:=v][x''=-g,t'=1\,\&\,x\geq 0\wedge v=v_0-(t-t_0)g]B_{(x,v)}$$

As this proof shows, everything works as expected as long as we realize that this requires a change of the invariants used for the differential cuts. The solution of the velocity to differentially cut in will be $v = v_0-(t-t_0)g$ and the solution of the position to differentially cut in subsequently will be $x = x_0+v_0(t-t_0)-\frac{g}{2}(t-t_0)^2$. With some thought you can also make sure to use the discrete ghosts for the initial values cleverly to initialize it at 0, which is significantly more convenient.

Note 63 (Ghost solutions) *Whenever there is a solution of a differential equation that we would like to make available to a proof without using the solution axiom schema [′], a differential cut and subsequent differential invariant can be used to cut the solution as an invariant into the system after a discrete ghost that remembers the initial values needed to express the solution. The tricky part is that solutions depend on time, and time may not be part of the differential equation system. If there is no time variable, however, an additional differential equation first needs to be added that pretends to be time.*

For the case of the bouncing ball, this proof looks unnecessarily complicated, because the solution axiom $[']$ could have been used instead right away, instead. Yet, even if this particular proof was more involved, the arithmetic ended up being nearly trivial in the end (which Note 61 on p. 336 already observed to hold in general for differential invariant proofs). But the same proof technique of adding ghost variables as needed can be pretty useful in more complicated systems.

Note 64 (On the utility of ghosts) *Adding ghosts as needed can be useful in more complicated systems that do not have computable solutions, but in which other relations between initial (or intermediate) and final state can be proved. The same technique can also be useful for cutting in solutions when only part of a differential equation system admits a polynomial solution.*

For example, the differential equation system $v_1' = \omega v_2, v_2' = -\omega v_1, v' = a, t' = 1$ is difficult, because it has non-polynomial solutions. Still, one part of this differential equation, the velocity $v' = a$, is easily solved. Yet, the solution axiom $[']$ is not applicable, because no real arithmetic solution of the whole differential equation system exists (except when $\omega = 0$). Regardless, after suitable discrete ghosts, a differential cut with the solution $v = v_0 + at$ of $v' = a$ adds this precise knowledge about the time-dependent change of the variable v to the evolution domain for subsequent use.

The ghost solution technique is a useful technique to prove properties of differential equations by using solutions that are first proved to be solutions (with a differential cut) and then used in the remaining proof (e.g. by differential weakening). Unlike the solution axiom schema $[']$, the ghost solution approach also works if only part of a differential equation system can be solved simply by cutting the required part of the solutions into the differential equations.

12.3.3 Differential Ghosts of Time

When we look back, the formula (12.2) that we now proved with ghost solutions had a differential equation $t' = 1$ for time that was not actually part of the original formula (11.7). Does that matter? Well, without having a time variable t, we could hardly have even meaningfully written down the solutions (12.1). That does not seem fair that this formula only has a proof by a differential cut with a solution if it already has a differential equation $t' = 1$ for time to begin with! Even if $t' = 1$ is not in the original differential equation, there should be a way to add it into the problem.

Indeed, come to think about it, every differential equation deserves a time variable if it needs one. It does not really change the system if we simply add a new differential equation for a time variable into it. Of course, we had better make sure that the variable is actually new and do not accidentally reuse a variable that was already present. For example, squeezing $x' = 1$ into the differential equation $x' = v, v' = -g$ for falling balls would seriously confuse the system dynamics, because x could hardly simultaneously follow $x' = v$ and the conflicting $x' = 1$. We

also cannot just squeeze in $g' = 1$ without significantly changing the dynamics of the system, because gravity g was supposed to be a constant in $x' = v, v' = -g$ and would suddenly be increasing over time in $x' = v, v' = -g, g' = 1$. But if we refrain from doing any of those silly mistakes and do not change the dynamics that was already there but merely add a dynamics that was not there yet, then adding a new time variable seems like a fine thing to do.

Now, if we want a way of adding a time variable into differential equation systems, we could add a proof rule just for exactly that purpose. The following proof rule makes it possible to add a new differential equation for time:

$$\frac{\Gamma \vdash [x' = f(x), t' = 1 \,\&\, Q]P, \Delta}{\Gamma \vdash [x' = f(x) \,\&\, Q]P, \Delta} \quad (t \text{ fresh}) \tag{12.4}$$

A soundness justification for this proof rule would use that, as long as t is a fresh variable that does not occur in the conclusion, then a proof of safety of the bigger differential equation system with $t' = 1$ implies safety of the differential equation without. Indeed, this proof rule could have been used to prove the original falling ball formula (11.7) from the above proof of (12.2). For once, the issue with this proof rule is not one of soundness, but rather a matter of economy of reasoning principles.

Proof rule (12.4) does a fine job adding clocks but cannot do anything else. If we ever want to add another differential equation into a differential equation system, this narrow-minded proof rule is of no use. And, in fact, the original proof rules for adding differential equations (called differential auxiliaries) allowed the addition of much more general differential equations [6] and were just subsequently also used for adding time variables as a special case. So before we end up wasting more time with the special case of time as a motivation, let's proceed right away the general case of differential ghosts, which are ghost variables with differential equations.

12.3.4 Constructing Differential Ghosts

Differential ghosts are ghost variables that are added into a differential equation system for the purpose of conducting a proof. The proof technique of differential ghosts is not limited to adding just the differential equation $t' = 1$ for time, but can add other differential equations $y' = g(x,y)$ into the differential equation system as well. In previous chapters, it has served us very well to first develop an axiomatic formulation and then proceed with packaging it up as the most useful proof rule subsequently. Let's proceed in the same way.

When we have a formula $[x' = f(x) \,\&\, Q]P$ about a differential equation (or a system) $x' = f(x) \,\&\, Q$, then we can add a new differential equation $y' = g(x,y)$ for a new variable y to obtain $[x' = f(x), y' = g(x,y) \,\&\, Q]P$. At what initial value does this new differential equation $y' = g(x,y)$ for the new differential ghost y start?

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

For the purposes of adding a differential ghost for a time variable $t' = 1$ for ghost solutions in Sect. 12.3.3 it would be best if the new variable were to start at 0. But for other use cases it might be much better to start the differential ghost elsewhere at a point that best fits to the subsequent proof. Does it matter for soundness where the differential ghost y starts?

Since the differential ghost y is a new variable that was not in the original question and is merely added for the sake of the argument, it can also start at any initial state that we like. That phenomenon is somewhat similar to discrete ghosts, which can also soundly assume any arbitrary initial value by rule IA. This calls for the use of an existential quantifier for the initial value of the differential ghost y, because any initial value would justify the original formula. And, in fact, also vice versa, the original formula implies the existence of an initial value for the ghost y for which the bigger differential equation system $x' = f(x), y' = g(x,y) \,\&\, Q$ always stays in P. These thoughts lead to the following formulation of the differential ghost axiom:

$$\text{DG}\quad [x' = f(x) \,\&\, Q]P \leftrightarrow \exists y\, [x' = f(x), y' = g(x,y) \,\&\, Q]P \tag{12.5}$$

Of course, y needs to be a new variable that does not occur in $[x' = f(x) \,\&\, Q]P$, since y is not much of a differential ghost if it was around before. Besides, it would be unsound to add a new differential equation for a variable that was used for a different purpose previously. If $x' = f(x) \,\&\, Q$ always stays in P, then there is an initial value for the differential ghost such that the augmented differential equation system $x' = f(x), y' = g(x,y) \,\&\, Q$ also always stays in P, and vice versa.

Certainly, the rule (12.4) for adding time can be derived from axiom DG when using 1 for $g(x,y)$. In fact, when cleverly instantiating by rule $\exists$R the resulting existential quantifier for the ghost with 0, even the following improved rule derives:

$$\frac{\Gamma, t = 0 \vdash [x' = f(x), t' = 1 \,\&\, Q]P, \Delta}{\Gamma \vdash [x' = f(x) \,\&\, Q]P, \Delta} \quad (t \text{ fresh})$$

This rule is more helpful for ghost solutions because it makes sure the differential equation actually starts at time $t = 0$, which simplifies the arithmetic considerably.

But other differential equations $y' = g(x,y)$ can be added by axiom DG as well. How general could they be? And what could they be good for? Is there a limit to what differential equations could be added?

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

At the latest during the soundness proof for axiom DG it will turn out that there is a limit to the differential equations that can be added soundly. But before proceeding with this crucial question, we take the inexcusable liberty of first exploring potential use cases of differential ghost axiom DG to sharpen our intuition and learn to appreciate what more general differential ghosts could even be good for.

Example 12.1 (Matters get worse without differential ghosts). A guiding example for the use of differential ghosts is the following simple formula:

$$x > 0 \rightarrow [x' = -x]\, x > 0 \tag{12.6}$$

This formula is not susceptible to differential invariance proofs using just the rule dI, because the trend along $x' = -x$ for the truth-value of the postcondition $x > 0$ makes matters worse over time (Fig. 12.1). The differential $-x \geq 0$ of $x > 0$ is invalid:

$$\text{dI}\,\dfrac{\text{[:=]}\,\dfrac{\dfrac{\text{not valid}}{\vdash -x \geq 0}}{\vdash [x' := -x]\, x' \geq 0}}{x > 0 \vdash [x' = -x]\, x > 0}$$

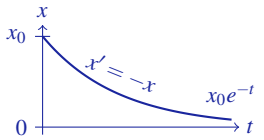


Fig. 12.1 Exponential decay along $x' = -x$ always makes matters worse for $x > 0$

With significantly more thought, it can be shown that there is no indirect way of making (12.6) provable with the help of differential cuts either [6]. But even if along $x' = -x$ the postcondition $x > 0$ tends toward becoming *false*, the rate at which it is becoming *false* is slowing down as well, because the differential equation is $x' = -x$ (and not $x' = -x - 1$ where the extra offset -1 would indeed ultimately make x negative). While $x > 0$ is making a race to the bottom, the rate at which x changes along $x' = -x$ simultaneously makes a race to the bottom (toward 0). That begs the question which of those two limit processes wins. If only we had a way to relate to an extra quantity that serves as a counterweight to the change of x and describes to what extent the rate at which x changes is being held up.

Suppose we had a differential ghost as an additional variable y with some differential equation that is still to be determined, which we can use as such a counterweight. What relationship of x and y would imply that x must be positive so that the postcondition $x > 0$ is *true*? If we think back of Lie's characterization of invariant terms in Sect. 10.8.3, then we recall that differential invariants are perfect at proving invariant terms that never change their value. The simple-most equation of x and y that implies $x > 0$ is $xy^2 = 1$, because y^2 is surely nonnegative and, thus, x must have been positive if its product with the nonnegative number y^2 is 1 (or any other positive number). And, in fact, $\exists y\, xy^2 = 1$ is even equivalent to $x > 0$.

Now, the only remaining question is by which differential equation the differential ghost y should change over time to preserve the invariant $xy^2 = 1$, which would imply the desired postcondition $x > 0$. This is the cool thing about differential ghosts: We get to choose their differential equations at our pleasure as $g(x,y)$ in axiom DG. Everywhere else do the variables change according to their very own

fixed differential equations that the original hybrid system comes with. But differential ghosts are different. We can make them change any way we want. We just need to make up our mind and choose cleverly to make our favorite proofs work out. How would we need y to change?

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

While it may sound like a big open question, it is completely systematic how we need a differential ghost to evolve if we want it to make a formula such as $xy^2 = 1$ an invariant. The formula will be an invariant if its differential $x'y^2 + x2yy' = 0$ proves along the differential equation. The differential equation already tells us that x' is $-x$, but we have not settled on our favorite differential equation for y' yet. Of course, the resulting formula $-xy^2 + 2xyy' = 0$ is best made *true* by a choice of $\frac{y}{2}$ for y', which we find by simply solving $-xy^2 + 2xyy' = 0$ for y'. That calls for the differential ghost $y' = \frac{y}{2}$ with which we set out for a proof of (12.6) using the dynamics illustrated in Fig. 12.2:

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{xy^2=1 \vdash x>0}\,\mathbb{R}}{x>0,xy^2=1 \vdash [x'=-x,y'=\frac{y}{2}]xy^2=1}\,\exists\mathrm{R}}{x>0 \vdash \exists y\,[x'=-x,y'=\frac{y}{2}]xy^2=1}\,\mathrm{MR}}{x>0 \vdash \exists y\,[x'=-x,y'=\frac{y}{2}]x>0}\,\mathrm{DG}$$

$$\dfrac{x>0 \vdash \exists y\,[x'=-x,y'=\frac{y}{2}]x>0}{x>0 \vdash [x'=-x]x>0}$$

$$\dfrac{\dfrac{*}{\vdash -xy^2+2xy\frac{y}{2}=0}\,\mathbb{R}}{\vdash [x':=-x][y':=\frac{y}{2}]x'y^2+x2yy'=0}\,[:=]\,\mathrm{dI}$$

The monotonicity step MR replaces the original postcondition $x > 0$ by the desired postcondition $xy^2 = 1$ that implies $x > 0$. Recall that $\exists y\, xy^2 = 1$ is equivalent to $x > 0$, which makes it particularly easy to see[3] why this monotonicity step also works in the context $\exists y$. From the desired equation $xy^2 = 1$ it would be easy to read off the concrete witness $\sqrt{\frac{1}{x}}$ for the existential quantifier instantiation step $\exists$R since $x > 0$ initially. But since existence of such a y is all that matters for the proof, it is enough to prove $\exists y\, xy^2 = 1$ from the antecedent $x > 0$, which is straightforward real arithmetic. Nice, this proof with a creative differential ghost gave us a surprising but systematic proof for exponential decay formula (12.6).

12.3.5 Differential Ghosts

Now that we saw the benefit of dreaming up an entirely new differential equation for a differential ghost for the sole purpose of doing a proof, it is about time to come back and see whether there are any limitations to the differential equations that can be added as differential ghosts for the purpose of a proof. What could possibly go

[3] Just like modalities and universal quantifiers, existential quantifiers also satisfy the monotonicity rule that $\exists y\, P \to \exists y\, Q$ can be derived from $P \to Q$.

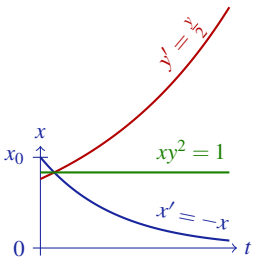


Fig. 12.2 Differential ghost y as counterweight for exponential decay along $x' = -x$

wrong? Because the new variables are really new, adding new variables with new differential equations should not affect the original differential equations, because they cannot even mention the differential ghosts.

The catch is that there is still a pretty subtle influence that additional differential equations can have on a preexisting differential equation system. If poorly chosen, then extra differential equations for differential ghosts could limit the duration of existence of the solution of the joint differential equation system. It would not help making a real system any safer if we were to compose it with a differential ghost that makes the imaginary world explode before the real system has a chance to run into an unsafe state.

Example 12.2 (Nonexistent differential ghosts). It would be unsound to add a differential ghost if its solution does not exist for at least as long as the original differential equation system has a solution. Otherwise the following unsound proof attempt would reduce a conclusion that is not valid to a premise that is valid by adding a differential ghost $y' = y^2+1$ whose solution $y(t) = \tan t$ does not even exist for long enough to make $x \leq 6$ *false* (as illustrated in Fig. 12.3):

$$\lightning \frac{x = 0, y = 0 \vdash [x' = 1, y' = y^2+1]\, x \leq 6}{\exists \mathrm{R}\ \dfrac{x = 0 \vdash \exists y\, [x' = 1, y' = y^2+1]\, x \leq 6}{x = 0 \vdash [x' = 1]\, x \leq 6}}$$

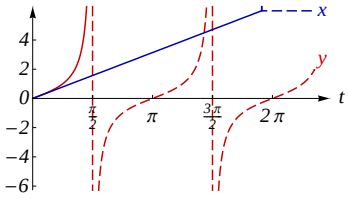


Fig. 12.3 Explosive differential ghosts that do not exist long enough would unsoundly limit the duration of solutions

When adding a differential ghost into a differential equation system it is, thus, crucial for soundness that the ghost differential equation has a solution that exists at

least as long as the solutions of the rest of the differential equation system exist. The easiest way of making that happen is if the new differential equation added for the differential ghost y is linear in y so of the form $y' = a(x) \cdot y + b(x)$. The terms $a(x)$ and $b(x)$ can be any terms of dL mentioning any number of variables any number of times, but they cannot mention y, because $y' = a(x) \cdot y + b(x)$ would not, otherwise, be linear in y. This leads to the following (soundness-critical) correction for the formulation of the differential ghost axiom from (16) that we initially suspected in Sect. 12.3.4.

Lemma 12.3 (Differential ghosts). *The differential ghost axiom DG is sound:*

$$\text{DG}\quad [x' = f(x)\,\&\,Q]P \leftrightarrow \exists y\,[x' = f(x), y' = a(x) \cdot y + b(x)\,\&\,Q]P$$

where y is a new variable, so neither occurs in the left-hand side $[x' = f(x)\,\&\,Q]P$ nor in $a(x)$ or $b(x)$.

Proof (Sketch). The proof, which is reported in full in prior work [8], uses a slight generalization of Corollary 2.1 from Sect. 2.9.2 to show that the new differential equation $y' = a(x) \cdot y + b(x)$ has a solution that exists at least as long as the solution of $x' = f(x)$ exists. The required Lipschitz condition follows from the fact that $a(x)$ and $b(x)$ in $y' = a(x) \cdot y + b(x)$ have continuous values over time, thus, assume their maximum on the compact interval of existence of the solution of x. □

Axiom DG can be used to show that a property P holds after a differential equation if and only if it holds for some initial value y after an augmented differential equation with an extra $y' = a(x) \cdot y + b(x)$ that is linear in y so still has a solution that exists sufficiently long. The case where $x' = f(x)$ is a (vectorial) differential equation system is accordingly, giving $y' = a(x) \cdot y + b(x)$ the opportunity to mention all variables other than the new y in $a(x)$ and $b(x)$.

A proof rule for differential ghosts derives from a direct use of axiom DG.

Lemma 12.4 (Differential ghost rule). *This proof rule derives from DG:*

$$\text{dG}\quad \frac{\Gamma \vdash \exists y\,[x' = f(x), y' = a(x) \cdot y + b(x)\,\&\,Q]P, \Delta}{\Gamma \vdash [x' = f(x)\,\&\,Q]P, \Delta} \quad (\textit{where } y \textit{ is new})$$

Proof. Proof rule dG derives by a straightforward application of axiom DG. □

As illustrated in Example 12.1, it is almost always beneficial to subsequently replace the postcondition P by a formula that makes use of the differential ghost y. The following rule DA was the first form [6] of differential ghosts and already bundles axiom DG up with others to a commonly useful form that adds a differential ghost while simultaneously replacing the postcondition to use the ghost.

Lemma 12.5 (Differential auxiliaries rule). *The* differential auxiliaries *proof rule for introducing new auxiliary differential variables y derives from DG:*

$$\text{DA}\ \frac{\vdash F \leftrightarrow \exists y\, G \quad \Gamma, G \vdash [x' = f(x), y' = a(x)\cdot y + b(x)\,\&\,Q]G, \Delta}{\Gamma, F \vdash [x' = f(x)\,\&\,Q]F, \Delta}$$

Proof. Rule DA derives from DG with transformations of the postcondition:

$$\text{DG}\ \frac{\text{MR}\ \dfrac{\dfrac{\exists y\, G \vdash F}{G \vdash F}\quad \exists\text{R,cut}\ \dfrac{F \vdash \exists y\, G \qquad \Gamma, G \vdash [x' = f(x), y' = a(x)\cdot y + b(x)]G, \Delta}{\Gamma, F \vdash \exists y\,[x' = f(x), y' = a(x)\cdot y + b(x)]G, \Delta}}{\Gamma, F \vdash \exists y\,[x' = f(x), y' = a(x)\cdot y + b(x)]F, \Delta}}{\Gamma, F \vdash [x' = f(x)]F, \Delta}$$

□

By the right premise of rule DA, for any y, G is an invariant of the extended dynamics. Thus, G always holds after the evolution for some y (its value can be different than in the initial state), which still implies F by the left premise. Since y is fresh and its linear differential equation does not limit the duration of solutions of x on Q, this implies the conclusion. Since y is fresh, y does not occur in Q, and, thus, its solution does not leave Q, which would incorrectly restrict the duration of the evolution.

Intuitively, rule DA can help proving properties, because it may be easier to characterize how x changes in relation to an auxiliary differential ghost variable y with a suitable differential equation ($y' = a(x)\cdot y + b(x)$) compared to understanding the change of x in isolation. As usual, it would not be sound to keep the context Γ, Δ on the first premise of rule DA, because we have no reason to believe it would still hold after the differential equation, where we only know G (for some current value of y) according to the second premise but need to conclude that F also holds.

We conclude this section with a series of instructive examples that convey how differential ghosts are used as a powerful proof technique for differential equations. In all examples is the differential equation for the differential ghost constructed entirely systematically as in Sect. 12.3.4 from the property that we want to be invariant.

Example 12.3 (Differential ghosts describe exponential growth). Exponential decay is not the only kind of dynamics that benefits from a differential ghost. Exponential growth is also proved just by flipping the sign of the differential ghost (Fig. 12.4).

$$\text{DA}\ \frac{\mathbb{R}\ \dfrac{*}{\vdash x > 0 \leftrightarrow \exists y\, xy^2 = 1} \qquad \dfrac{\mathbb{R}\ \dfrac{*}{\vdash xy^2 + 2xy(-\frac{y}{2}) = 0}}{[:=]\ \dfrac{\vdash [x':=x][y':=-\frac{y}{2}]x'y^2 + x2yy' = 0}{\text{dI}\ \dfrac{}{xy^2 = 1 \vdash [x' = x, y' = -\frac{y}{2}]xy^2 = 1}}}}{x > 0 \vdash [x' = x]x > 0}$$

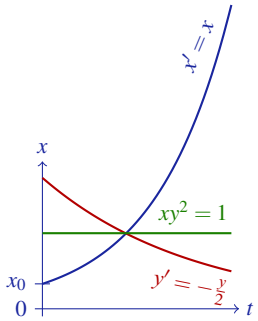


Fig. 12.4 Differential ghost y to balance for exponential growth along $x' = x$

Example 12.4 (Exponential difference). It is equally easy to prove that x will never become zero along an exponential decay $x' = -x$. In this case, the condition of x and the additional differential ghost y that is equivalent to $x \neq 0$ is $\exists y\, xy = 1$, which requires the slightly different differential equation $y' = y$ to become invariant:

$$\text{DA}\,\dfrac{\mathbb{R}\,\dfrac{*}{\vdash x > 0 \leftrightarrow \exists y\, xy = 1} \qquad \text{dI}\,\dfrac{[:=]\,\dfrac{\mathbb{R}\,\dfrac{*}{\vdash -xy + xy = 0}}{\vdash [x' := -x][y' := y]x'y + xy' = 0}}{xy = 1 \vdash [x' = -x, y' = y]\, xy = 1}}{x \neq 0 \vdash [x' = -x]\, x \neq 0}$$

Example 12.5 (Weak exponential decay). Proving that $x \geq 0$ is an invariant for exponential decay $x' = -x$ results in two cases, the case where $x = 0$ and where $x > 0$. Distinguishing between both cases results in a successful proof. It is easier, however, to keep both cases together by rephrasing the desired invariant $x \geq 0$ with the equivalent $\exists y\,(y > 0 \wedge xy \geq 0)$ using a corresponding differential ghost $y' = y$.

$$\text{DA}\,\dfrac{\mathbb{R}\,\dfrac{*}{\vdash x \geq 0 \leftrightarrow \exists y\,(y > 0 \wedge xy \geq 0)} \qquad \text{dI}\,\dfrac{[:=]\,\dfrac{\mathbb{R}\,\dfrac{*}{\vdash -xy + xy \geq 0}}{\vdash [x' := -x][y' := y]x'y + xy' \geq 0}}{xy \geq 0 \vdash [x' = -x, y' = y]\, xy \geq 0}\,\triangleleft \qquad []\wedge\,\dfrac{}{y > 0 \wedge xy \geq 0 \vdash [x' = -x, y' = y](y > 0 \wedge xy \geq 0)}}{x \geq 0 \vdash [x' = -x]\, x \geq 0}$$

The $[]\wedge$ derived axiom leads to another premise (marked by $\triangleleft$), which proves with yet another differential ghost just like in Example 12.3, just carrying $x' = -x$ around:

$$\text{DA}\dfrac{\mathbb{R}\dfrac{*}{\vdash y>0\leftrightarrow\exists z\, yz^2=1}\qquad \begin{array}{l}[:=]\dfrac{\mathbb{R}\dfrac{*}{\vdash yz^2+2yz(-\frac{z}{2})=0}}{\vdash [y':=y][z':=-\frac{z}{2}]y'z^2+y2zz'=0}\\ \text{dI}\dfrac{}{yz^2=1\vdash[x'=-x,y'=y,z'=-\frac{z}{2}]yz^2=1}\end{array}}{y>0\vdash[x'=-x,y'=y]\,y>0}$$

Example 12.6 (Exponential equilibrium). Proving that $x=0$ is an invariant for exponential decay $x'=-x$ succeeds by rephrasing it with the equivalent invariant $\exists y\,(y>0\land xy=0)$ and then following an analogue of Example 12.5. Alternatively, the splitting axiom $[]\land$ can directly reuse the proof of Example 12.5 to show that both $x\geq 0$ and $x\leq 0$ are invariants of $x'=-x$, which imply that their conjunction $x=0$ is invariant as well.

12.3.6 Substitute Ghosts

Differential ghosts even give us a, shockingly spooky, way of generating differential equations for differential ghosts on the fly as needed for proofs to work out. That might sound scary but is amazingly useful. To see how it works, invent your own differential ghost $y'=$ ☁ with a still-unspecified right-hand side ☁, which is nothing but a substitute ghost or a common spooky cloud, and just keep "proving" as if nothing had happened:

$$\text{DA}\dfrac{\mathbb{R}\dfrac{*}{\vdash x>0\leftrightarrow\exists y\, xy^2=1}\qquad \begin{array}{l}[:=]\dfrac{\dfrac{\text{could prove if } ☁=\frac{y}{2}}{\vdash -xy^2+2xy☁=0}}{\vdash [x':=-x][y':=☁]x'y^2+x2yy'=0}\\ \text{dI}\dfrac{}{xy^2=1\vdash[x'=-x,y'=☁]xy^2=1}\end{array}}{x>0\vdash[x'=-x]x>0}$$

The right premise could prove if only ☁ were chosen to be $\frac{y}{2}$, in which case the premise $-xy^2+2xy☁=0$ is quite easily proved. That, of course, was a bit too spooky for the soundness-loving truth-connoisseur. So let's instantiate the spooky cloud ☁ with its concrete choice $\frac{y}{2}$ and start all over with a proper proof:

$$\text{DA}\dfrac{\mathbb{R}\dfrac{*}{\vdash x>0\leftrightarrow\exists y\, xy^2=1}\qquad \begin{array}{l}[:=]\dfrac{\mathbb{R}\dfrac{*}{\vdash -xy^2+2xy\frac{y}{2}=0}}{\vdash [x':=-x][y':=\frac{y}{2}]x'y^2+x2yy'=0}\\ \text{dI}\dfrac{}{xy^2=1\vdash[x'=-x,y'=\frac{y}{2}]xy^2=1}\end{array}}{x>0\vdash[x'=-x]\,x>0}$$

Fortunately, this proper dL proof confirms the suspicion of a proof that we developed above. In that sense, all is fair in how we come up with a proof, even if we use spooky ghost arguments involving  .[4] But in the end, it is crucial to conduct a proper proof with sound proof rules to ensure the conclusion is valid.

It can be shown [6] that there are properties such as this one that crucially need differential ghosts (alias differential auxiliaries) to prove, which makes differential ghosts a powerful proof technique.

12.3.7 Axiomatic Ghosts

This section is devoted to yet another kind of ghosts: axiomatic ghosts. While less important for simple systems, axiomatic ghosts are the way to go for systems that involve special functions such as $\sin, \cos, \tan$ etc.

When neglecting wind, gravitation, and so on, which is argued to be appropriate for analyzing cooperation in air traffic control [10], the in-flight dynamics of an aircraft at x can be described by the following differential equation system; see [10] for details:

$$x_1' = v\cos\vartheta \qquad x_2' = v\sin\vartheta \qquad \vartheta' = \omega \tag{12.7}$$

That is, the linear velocity v of the aircraft changes both positions x_1 and x_2 in the (planar) direction corresponding to the orientation ϑ the aircraft is currently heading toward. Further, the angular velocity ω of the aircraft changes the orientation ϑ of the aircraft.

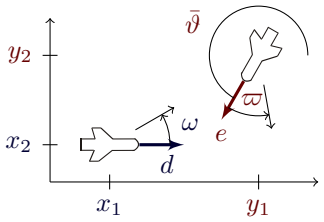


Fig. 12.5 Aircraft dynamics illustration

Unlike for straight-line flight ($\omega = 0$), the nonlinear dynamics in (12.7) is difficult to analyse [10] for curved flight ($\omega \neq 0$), especially due to the trigonometric expressions which are generally undecidable. Solving (12.7) requires the Floquet theory of differential equations with periodic coefficients [11, Theorem 18.X] and yields mixed polynomial expressions with multiple trigonometric functions. A true challenge, however, is the need to verify properties of the states that the aircraft reach by following these solutions, which requires proving that complicated formu-

[4] Of course,  is not quite as spooky as one might suspect. It can be made rigorous with function symbols that are subsequently substituted uniformly [8].

las with mixed polynomial arithmetic and trigonometric functions hold true for all values of state variables and all possible evolution durations. However, quantified arithmetic with trigonometric functions is undecidable by Gödel's incompleteness theorem [2].

To obtain polynomial dynamics, we axiomatize the trigonometric functions in the dynamics differentially and reparametrize the state correspondingly. Instead of angular orientation ϑ and linear velocity v, we use the linear speed vector

$$(v_1, v_2) \overset{\text{def}}{=} (v\cos\vartheta, v\sin\vartheta) \in \mathbb{R}^2$$

which describes both the linear speed $\|(v_1,v_2)\| := \sqrt{v_1^2+v_2^2} = v$ and the orientation of the aircraft in space; see Figs. 12.5 and 12.6. Substituting this coordinate

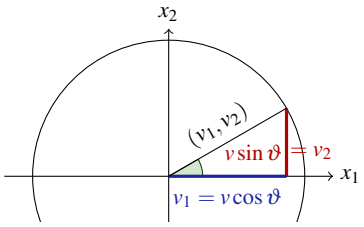


Fig. 12.6 Reparametrize for differential axiomatization

change into differential equations (12.7), we immediately have $x_1' = v_1$ and $x_2' = v_2$. With the coordinate change, we further obtain differential equations for v_1, v_2 from differential equation system (12.7) by simple symbolic differentiation:

$$\begin{aligned} v_1' &= (v\cos\vartheta)' = v'\cos\vartheta + v(-\sin\vartheta)\vartheta' = -(v\sin\vartheta)\omega &&= -\omega v_2 \\ v_2' &= (v\sin\vartheta)' = v'\sin\vartheta + v(\cos\vartheta)\vartheta' &&= (v\cos\vartheta)\omega = \omega v_1 \end{aligned}$$

The middle equality holds for constant linear velocity ($v' = 0$), which we assume, because only limited variations in linear speed are possible and cost-effective during the flight [3, 10] so that angular velocity ω is the primary control parameter in air traffic control. Hence, equations (12.7) can be restated as the following differential equation:

$$x_1' = v_1\,,\, x_2' = v_2\,,\, v_1' = -\omega v_2\,,\, v_2' = \omega v_1 \tag{12.8}$$

$$y_1' = e_1\,,\, y_2' = u_2\,,\, u_1' = -\rho u_2\,,\, u_2' = \rho u_1 \tag{12.9}$$

Differential equation (12.8) expresses that position $x = (x_1, x_2)$ changes according to the linear speed vector (v_1, v_2), which in turn rotates according to ω. Simultaneous movement together with a second aircraft at $y \in \mathbb{R}^2$ having linear speed $(u_1, u_2) \in \mathbb{R}^2$ (also indicated with angle $\bar{\vartheta}$ in Fig. 12.5) and angular velocity ρ corresponds to the differential equation system (12.8) together with (12.9). Differential equations capture simultaneous dynamics of multiple traffic agents succinctly using conjunction.

By this *differential axiomatization*, we thus obtain polynomial differential equations. Note, however, that their solutions still involve the same complicated nonlinear trigonometric expressions so that solutions still give undecidable arithmetic [5, Appendix B]. Note that differential invariant type arguments work with the differential equations themselves and not with their solutions, so that differential axiomatization actually helps proving properties, because the solutions are still as complicated as they have always been, but the differential equations become easier

The same technique helps when handling other special functions in other cases by differential axiomatization.

12.4 Summary

The major lesson from this chapter is that it can sometimes be easier to relate a variable to its initial value or to other quantities than to understand its value in isolation. Ghosts, in their various forms, let us achieve that by adding auxiliary variables into the system dynamics, so that the values of the original variables of interest can be related to the values of the ghosts. Sometimes such ghosts are even necessary to prove properties. As a workaround, it might help to rewrite the original model so that it already includes the ghost variables, preferably marked as ghosts in the model. The phenomenon that relations between state and ghost variables are sometimes easier to prove than just standalone properties of state variables applies in either case. This chapter shines a light on the power of relativity theory in the sense of relating variables to one another.

Lemma 12.3 (Differential ghosts). *The differential ghost axiom DG is sound:*

$$\text{DG}\quad [x' = f(x)\,\&\,Q]P \leftrightarrow \exists y\,[x' = f(x), y' = a(x)\cdot y + b(x)\,\&\,Q]P$$

where y is a new variable, so neither occurs in the left-hand side $[x' = f(x)\,\&\,Q]P$ nor in $a(x)$ or $b(x)$.

Lemma 12.4 (Differential ghost rule). *This proof rule derives from DG:*

$$\text{dG}\quad \frac{\Gamma \vdash \exists y\,[x' = f(x), y' = a(x)\cdot y + b(x)\,\&\,Q]P, \Delta}{\Gamma \vdash [x' = f(x)\,\&\,Q]P, \Delta} \quad (\textit{where } y \textit{ is new})$$

This chapter also showcased a number of other useful proof techniques and even showed how properties of differential equations can be proved using solution-like arguments if only part of the differential equation system can be solved.

12.5 Appendix

12.5.1 Arithmetic Ghosts

The easiest way to see why it sometimes makes sense to add variables into a system model is to take a look at divisions. Divisions are not officially part of real arithmetic, because divisions can be defined. For example, when a division b/c is ever mentioned in a term such as $q = b/c$, then we can characterize q to remember the value of b/c by indirectly characterizing q in terms of b and c without $/$ and then subsequently use q wherever b/c first occurred:

$$q := \frac{b}{c} \quad \rightsquigarrow \quad q := *;\ ?qc = b \quad \rightsquigarrow \quad q := *;\ ?qc = b \wedge c \neq 0$$

where $q := *$ is the nondeterministic assignment that assigns an arbitrary real number to q. The first transformation (simply written $\rightsquigarrow$) characterizes $q = b/s$ indirectly by multiplying up as $qc = b$. The second transformation then conscientiously remembers that divisions only make sense when we avoid dividing by zero. After all, divisions by zero excel at causing a lot of trouble. This transformation can be used when b/c occurs in the middle of a term too:

$$x := 2 + \frac{b}{c} + e \rightsquigarrow q := *;\ ?qc = b;\ x := 2 + q + e \rightsquigarrow q := *;\ ?qc = b \wedge c \neq 0;\ x := 2 + q + e$$

Here q is called an *arithmetic ghost*, because q is an auxiliary variable that is only added to the hybrid program for the sake of defining the arithmetic quotient $\frac{b}{c}$. In similar ways can we define other functions like square roots using an arithmetic ghost:

$$x := a + \sqrt{4y} \quad \rightsquigarrow \quad q := *;\ ?q^2 = 4y;\ x := a + q$$

But we should again scrutinize to make sure we realize that $4y$ should be nonnegative for the square root to make sense and could indeed add that into the test. We settle on not doing so, since non-negativity already follows from $q^2 = 4y$.

12.5.2 Nondeterministic Assignments & Ghosts of Choice

The HP statement $x := *$ that has been used in Sect. 12.5.1 is a nondeterministic assignment that assigns an arbitrary real number to x. Comparing with the syntax of hybrid programs from Chap. 3, however, it turns out that such a statement is not in the official language of hybrid programs.

$$\alpha, \beta \ ::= \ x := e \mid ?Q \mid x' = f(x)\,\&\,Q \mid \alpha \cup \beta \mid \alpha;\beta \mid \alpha^* \tag{12.10}$$

What now?

One possible solution, which is the one taken in the implementation of the hybrid systems theorem prover KeYmaera [9] and its successor KeYmaera X [1], is to add the nondeterministic assignment $x := *$ as a statement to the syntax of hybrid programs.

$$\alpha, \beta \ ::= \ x := e \mid ?Q \mid x' = f(x) \,\&\, Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid x := *$$

Consequently, the new syntactic construct of nondeterministic assignments needs a semantics to become meaningful:

7. $[\![x := *]\!] = \{(\omega, \nu) \ : \ \nu = \omega$ except for the value of x, which can be any real$\}$

Then nondeterministic assignments also need axioms or proof rules so that they can be understood and analyzed in proofs. Those are reported in Fig. 12.7.

$\langle :* \rangle \ \langle x := * \rangle P \leftrightarrow \exists x P$

$[:*] \ [x := *] P \leftrightarrow \forall x P$

Fig. 12.7 Axioms for nondeterministic assignments

Axiom $\langle :* \rangle$ says that there is one way of assigning an arbitrary value to x so that P holds afterwards (i.e. $\langle x := * \rangle P$ holds) if and only if P holds for some value of x (i.e. $\exists x P$ holds). And axiom $[:*]$ says that P holds for all ways of assigning an arbitrary value to x (i.e. $[x := *] P$ holds) if and only if P holds for all values of x (i.e. $\forall x P$ holds), because x might have any such value after running $x := *$, and because the $[\alpha]$ means that the postcondition needs to be true after all ways of running α.

An alternative approach for adding nondeterministic assignments $x := *$ to hybrid programs is to reconsider whether we even have to do add a new construct for $x := *$ or whether it can already be expressed in other ways. That is, to understand whether $x := *$ is truly a new program construct or whether it can be defined in terms of the other hybrid program statements from (12.10). Is $x := *$ definable by another hybrid program?

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

According to the proof rules $[:*]$ and $\langle :* \rangle$, nondeterministic assignments $x := *$ can be expressed equivalently by suitable quantifiers. But that does not help at all in the middle of a program, where we can hardly write down a quantifier to express that the value of x now changes.

There is another way, though. Nondeterministic assignment $x := *$ assigns any real number to x. One hybrid program that has the same effect of giving x any arbitrary real value [5, Chapter 3] is:[5]

[5] Observe a subtlety that, unlike the nondeterministic assignment, the differential equations also have an impact on the value of x', which is fine since most programs do not read x' any further, but needs extra care with an additional discrete ghost z otherwise: $z := x'; \{x' = 1 \cup x' = -1\}; [x' := z]$

$$x:=* \quad \stackrel{\text{def}}{\equiv} \quad x'=1 \cup x'=-1 \tag{12.11}$$

That is not the only definition of $x:=*$, though. An equivalent definition is [7]:

$$x:=* \quad \stackrel{\text{def}}{\equiv} \quad x'=1;\ x'=-1$$

When working through the intended semantics of the left-hand side $x:=*$ shown in Case 7 above and the actual semantics of the right-hand side of (12.11) according to Chap. 3, it becomes clear that both sides of (12.11) mean the same, because they have the same reachability relation. Hence, the above definition (12.11) captures the intended concept of giving x any arbitrary real value, nondeterministically. And, in particular, just like if-then-else, nondeterministic assignments do not really have to be added to the language of hybrid programs, because they can already be defined. Likewise, no proof rules would have to be added for nondeterministic assignments, because there are already proof rules for the constructs used in the right-hand side of the definition of $x:=*$ in (12.11). Since the above proof rules $\langle:*\rangle$,$[:*]$ for $x:=*$ are particularly easy, though, it is usually more efficient to include them directly, which is what KeYmaera does.

What may, at first sight, appear slightly spooky about (12.11), however, is that the left-hand side $x:=*$ is clearly an instant change in time where x changes its value instantaneously to some arbitrary new real number. That is not quite the case for the right-hand side of (12.11), which involves two differential equations, which take time to follow.

The clue is that this passage of time is not observable in the state of the system. Consequently, the left-hand side of (12.11) really means the same as the right-hand side of (12.11). Remember from earlier chapters that time is not special. If a CPS wants to refer to time, it would have a clock variable t with the differential equation $t'=1$. With such an addition, however, the passage of time t becomes observable in the value of variable t and, hence, a corresponding variation of the right-hand side of (12.11) would not be equivalent to $x:=*$ at all (indicated by $\not\equiv$):

$$x:=* \quad \not\equiv \quad x'=1,t'=1 \cup x'=-1,t'=1$$

Both sides differ, because the right side exposes the amount of time t it took to get the value of x to where it should be, which, secretly, records information about the absolute value of the change that x underwent from its old to its new value. That change is something that the left-hand side $x:=*$ knows nothing about.

12.5.3 Differential-algebraic Ghosts

The transformation in Sect. 12.5.1 can eliminate all divisions, not just in assignments, but also in tests and all other hybrid programs, with the notable exception of

differential equations. Eliminating divisions in differential equations turns out to be a little more involved.

The following elimination using a (discrete) arithmetic ghost q is correct:

$$x' = \frac{2x}{c} \,\&\, c \neq 0 \wedge \frac{x+1}{c} > 0 \quad \rightsquigarrow \quad q := *; ?qc = 1; \, \{x' = 2xq \,\&\, c \neq 0 \wedge (x+1)q > 0\}$$

where the extra ghost variable q is supposed to remember the value of $\frac{1}{c}$.

The following attempt with a (discrete) arithmetic ghost q, however, would change the semantics rather radically:

$$x' = \frac{c}{2x} \,\&\, 2x \neq 0 \wedge \frac{c}{2x} > 0 \quad \not\rightsquigarrow \quad q := *; ?q2x = 1; \, \{x' = cq \,\&\, 2x \neq 0 \wedge cq > 0\}$$

because q then only remembers the inverse of the initial value of $2x$, not the inverse of the value of $2x$ as x evolves along the differential equation $x' = \frac{c}{2x}$. That is q has a constant value during the differential equation but, of course, the quotient $\frac{c}{2x}$ changes over time since x does.

One way to proceed is to figure out how the value of the quotient $q = \frac{1}{2x}$ changes over time as x changes by $x' = \frac{c}{2x}$. By deriving what q stands for, that results in

$$q' = \left(\frac{1}{2x}\right)' = \frac{-2x'}{4x^2} = \frac{-2\frac{c}{2x}}{4x^2} = -\frac{c}{4x^3}$$

Alas, we go unlucky here, because that gives yet another division to keep track of.

The other and entirely systematic way to proceed is to lift nondeterministic assignments q to differential equations $q' = *$ with the intended semantics that q changes arbitrarily over time while following that nondeterministic differential equation:[6]

$$q' = \frac{b}{c} \quad \rightsquigarrow \quad q' = * \,\&\, qc = b \quad \rightsquigarrow \quad q' = * \,\&\, qc = b \wedge c \neq 0$$

While it is slightly more complicated to give a semantics to $q' = *$, the idea behind the transformation is completely analogous to the case of discrete arithmetic ghosts:

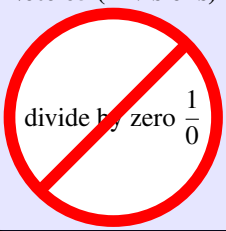
$$x' = 2 + \frac{b}{c} + e \quad \rightsquigarrow \quad x' = 2 + q + e, q' = * \,\&\, qc = b \quad \rightsquigarrow \quad x' = 2 + q + e, q' = * \,\&\, qc = b \wedge c \neq 0$$

Variable q is a *differential-algebraic ghost* in the sense of being an auxiliary variable in the differential-algebraic equation for the sake of defining the quotient $\frac{b}{c}$.

[6] See [5, Chapter 3] for the precise meaning of the nondeterministic differential equation $q' = *$. It is the same as the differential-algebraic constraint $\exists d \, q' = d$, but differential-algebraic constraints have not been introduced in this textbook so far, either. The intuition of allowing arbitrary changes of the value of q over time is fine, though, for our purposes.

Together with the reduction of divisions in discrete assignments from Sect. 12.5.1, plus the insight that divisions in tests and evolution domain constraints can always be rewritten to division-free form, this gives a (rather sketchy) proof showing that hybrid programs and differential dynamic logic do not need divisions [5]. The advantage of eliminating divisions this way is that differential dynamic logic does not need special precautions for divisions and that the handling of zero divisors is made explicit in the way the divisions are eliminated from the formulas. In practice, however, it is still useful to use divisions, yet great care has to be exercised to make sure that no inadvertent divisions by zero could ever cause troublesome singularities.

Note 65 (Divisions)



Whenever dividing, exercise great care not to accidentally divide by zero, for that will cause quite some trouble. More often than not, this trouble corresponds to missing requirements in the system. For example $\frac{v^2}{2b}$ *may be a good stopping distance when braking to a stop from initial velocity* v*, except when* $b = 0$*, which corresponds to having no brakes at all.*

Exercises

12.1 (Conditions for discrete ghosts). Identify a minimal set of conditions are necessary for proof rule IA from Sect. 12.3.1 to be sound. Show a counterexample for each of the remaining conditions to illustrate why it is necessary.

12.2. Augment the discrete ghost proofs in Sect. 12.3.1 to a full sequent proof of

$$xy - 1 = 0 \rightarrow [x' = x, y' = -y]xy = 1$$

12.3. Augment the proofs in this chapter as described to obtain a full sequent proof of (11.7). Be advised to find a big sheet of paper, first.

References

1. Fulton, N., Mitsch, S., Quesel, J.-D., Völp, M. & Platzer, A. *KeYmaera X: An Axiomatic Tactical Theorem Prover for Hybrid Systems* in *CADE* (eds Felty, A. & Middeldorp, A.) **9195** (Springer, 2015), 527–538. doi:10.1007/978-3-319-21401-6_36.
2. Gödel, K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Mon. hefte Math. Phys.* **38,** 173–198 (1931).

3. Livadas, C., Lygeros, J. & Lynch, N. A. High-Level Modeling and Analysis of TCAS. *Proc. IEEE - Special Issue on Hybrid Systems: Theory & Applications* **88,** 926–947 (2000).
4. Platzer, A. Differential-Algebraic Dynamic Logic for Differential-Algebraic Programs. *J. Log. Comput.* **20,** 309–352 (2010).
5. Platzer, A. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics* doi:10.1007/978-3-642-14509-4 (Springer, Heidelberg, 2010).
6. Platzer, A. The Structure of Differential Invariants and Differential Cut Elimination. *Log. Meth. Comput. Sci.* **8,** 1–38 (2012).
7. Platzer, A. Differential Game Logic. *ACM Trans. Comput. Log.* **17,** 1:1–1:51 (2015).
8. Platzer, A. A Complete Uniform Substitution Calculus for Differential Dynamic Logic. *J. Autom. Reas.* doi:10.1007/s10817-016-9385-1 (2016).
9. Platzer, A. & Quesel, J.-D. *KeYmaera: A Hybrid Theorem Prover for Hybrid Systems.* in *IJCAR* (eds Armando, A., Baumgartner, P. & Dowek, G.) **5195** (Springer, 2008), 171–178. doi:10.1007/978-3-540-71070-7_15.
10. Tomlin, C., Pappas, G. J. & Sastry, S. Conflict resolution for air traffic management: a study in multi-agent hybrid systems. *IEEE T. Automat. Contr.* **43,** 509–521 (1998).
11. Walter, W. *Ordinary Differential Equations* doi:10.1007/978-1-4612-0601-9 (Springer, 1998).